NTIA Technical Report TR-09-457

Techniques for Evaluating Objective Video Quality Models Using Overlapping Subjective Data Sets

Margaret H. Pinson
Stephen Wolf

report series

U.S. DEPARTMENT OF COMMERCE • National Telecommunications and Information Administration

NTIA Technical Report TR-09-457

Techniques for Evaluating Objective Video Quality Models Using Overlapping Subjective Data Sets

Margaret H. Pinson
Stephen Wolf

U.S. DEPARTMENT OF COMMERCE
Carlos M. Gutierrez, Secretary

Meredith A. Baker, Acting Assistant Secretary
for Communications and Information

November 2008

DISCLAIMER

This report presents supplemental data analyses of the Video Quality Experts Group (VQEG) Multi-Media (MM) Phase I experimental data. These supplemental data analyses were not submitted to VQEG for approval nor are they included in the VQEG MM Phase I final report that was submitted to various standards organizations.

This report evaluates objective video quality models that were submitted to VQEG in the MM Phase I validation tests. Models and their owners are identified in this report to specify adequately the technical aspects of the reported results. Certain commercial software are identified in this report to specify adequately the technical aspects of the reported results. In no case does such identification imply recommendation or endorsement by the National Telecommunications and Information Administration (NTIA), nor does it imply that the models or software identified are necessarily the best available for the particular application or use.

This document contains software developed by NTIA. **NTIA does not make any warranty of any kind, express, implied or statutory, including, without limitation, the implied warranty of merchantability, fitness for a particular purpose, non-infringement and data accuracy.** NTIA does not warrant or make any representations regarding the use of the software or the results thereof, including but not limited to the correctness, accuracy, reliability or usefulness of the software or the results. You can use, copy, modify and redistribute the NTIA-developed software in Appendix B upon your acceptance of these terms and conditions and upon your express agreement to provide appropriate acknowledgments of NTIA’s ownership of and development of the software by keeping this exact text present in any copied or derivative works.

CONTENTS

Page

FIGURES

Page

TABLES

Page

ABBREVIATIONS/ACRONYMS

ACR	Absolute Category Rating
ACR-HR	ACR with Hidden Reference
CI	Confidence Interval
CIF	Common Intermediate Format (352 by 288, square pixels)
DMOS	Differential Mean Opinion Score
DOC	Department of Commerce
FR	Full Reference
HRC	Hypothetical Reference Circuit
ITS	Institute for Telecommunication Sciences
MM	Multi-media
MPEG	Motion Picture Experts Group
NR	No Reference
NTIA	National Telecommunications and Information Administration
OR	Outlier Ratio
PSNR	Peak Signal-to-Noise Ratio
PVS	Processed Video Sequence
QCIF	Quarter CIF (176 by 144, square pixels)
RMSE	Root Mean Squared Error
RP	Resolving Power
RR	Reduced Reference
RV	Real Video
VC	Video Codec
VC-1	Video Codec 1, also known as Windows Media 9
VGA	Video Graphics Array (640 by 480, square pixels)
VM	Video Metric
VQEG	Video Quality Experts Group

EXECUTIVE SUMMARY

This report presents techniques for evaluating objective video quality models using overlapping subjective data sets. The techniques are demonstrated using data from the Video Quality Experts Group (VQEG) Multi-Media (MM) Phase I validation tests. These results also provide a supplemental analysis of the performance achieved by the objective models that were submitted to VQEG MM Phase I.

The VQEG MM Phase I primary analysis [1] provides confidence intervals that can be used to determine whether models are significantly different on a per-experiment basis. The problem is that there are 13 or 14 individual experiments at each image resolution (QCIF, CIF, and VGA), where each experiment has a different mix of source scenes, codecs, transmission errors, and other quality testing characteristics. Thus, each experiment yields a unique result for relative and absolute performance of the various models. Averaging the primary analysis statistics reduces the amount of data presented, but makes statistical significance testing difficult to compute. Summing the number of times a model is in the group of top-performing models has other problems and issues. Here, significance can be computed but the relative accuracy is lost.

Therefore, we chose to examine the MM data in a different fashion. The MM data consists of 41 individual experiments performed by many different laboratories throughout the world. A small common set of 30 video sequences (at each image resolution) was inserted into every subjective experiment. The approach presented herein was motivated by the very high laboratory-to-laboratory correlations of the subjective scores for this common set, and the fact that this common set spanned the full range of video quality that was presented in the subjective experiments. We used the common set at each image resolution to map all the subjective scores for all the experiments at that resolution onto a single subjective scale. This produces three supersets of subjective scores: QCIF, CIF, and VGA.

The three supersets produce powerful results that draw upon all of the video clips simultaneously, and allow us to delve into deeper questions such as the response of a model to specific coding algorithms and transmission errors, and the response of a model when one averages results from multiple scenes (from each video system under test). These results provide more detailed characterizations of each model and its comparative response to different stimuli (e.g., how a model's performance on coding-only impairments compares to its performance on transmission-error impairments). The subjective data supersets also allow us to compute new powerful statistical measures of model performance such as resolving power [2] [3] [4]. The resolving power of each model provides end-users an understanding of the precision supplied by their measurements.

Of the three metrics used by VQEG – Pearson correlation, Root Mean Squared Error (RMSE), and outlier ratio – RMSE is used most commonly in this report. RMSE provides the best discrimination and most flexible comparisons. Also of interest are comparisons that could not be made as a result of limitations in the experimental designs. This information may help researchers design future experiments.

TECHNIQUES FOR EVALUATING OBJECTIVE VIDEO QUALITY MODELS USING OVERLAPPING SUBJECTIVE DATA SETS

Margaret H. Pinson and Stephen Wolf [1]

This report presents techniques for evaluating objective video quality models using overlapping subjective data sets. The techniques are demonstrated using data from the Video Quality Experts Group (VQEG) Multi-Media (MM) Phase I experiments. These results also provide a supplemental analysis of the performance achieved by the objective models that were submitted to the MM Phase I experiments. The analysis presented herein uses the subjective scores from the common set of video clips to map all the subjective scores from the 13 or 14 experiments (at a given image resolution) onto a single subjective scale. This mapping greatly increases the available data and thus allows for more powerful analysis techniques to be performed. Resolving power values are presented for each model and image resolution. On a per-clip level, models' responses to stimuli are analyzed with respect to all stimuli, each coding algorithm, coding-only impairments, and transmission error impairments. The models' responses to stimuli are also analyzed on per-system and per-scene levels. Results indicate the amount of improvement possible when averaging over multiple scenes or systems.

Key words: combining; correlation; mapping; multi-media; objective; performance; quality; subjective; video; VQEG

1 INTRODUCTION

This report presents techniques for evaluating objective video quality models using overlapping subjective data sets. The techniques are demonstrated using data from the Video Quality Experts Group (VQEG) Multi-Media (MM) Phase I validation tests. These results also provide a supplemental analysis of the performance achieved by the objective models that were submitted to VQEG MM Phase I. The supplemental analysis procedure outlined in this document provides a unique insight into the relative and expected performance of the objective MM video quality models.

Section 2 presents a brief summary of the VQEG MM Phase I experiments while Section 3 presents an overview of our data analysis approach. Section 4 presents the results of the algorithm used to map subjective scores from the many individual experiments performed at each image resolution (QCIF, CIF, VGA) onto a single subjective scale for that image resolution. Section 5 presents the mapping function for each model, resolving power values for different levels of confidence, and values for the three performance metrics specified in the VQEG MM Phase I final report [1], namely Pearson correlation, Root Mean Squared Error (RMSE), and outlier ratio (OR). Section 6 presents statistics that analyze each model's response to different

[1] The authors are with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80305.

impairment types (e.g., specific codecs, coding-only impairments, transmission errors). Section 7 examines the impact on model accuracy when averaging over increasing numbers of source scenes to obtain improved system quality estimates.

2 SUMMARY OF THE VQEG MULTIMEDIA PHASE I EXPERIMENTS

The VQEG MM Phase I final report [1] describes the subjective experimental designs in great detail. A paraphrased summary from that report is given here to provide the reader with the necessary background to understand the data analysis that will be presented later in this report.

- The MM experiment examined video suitable for mobile/PDA and broadband internet communications services. The intent is that this video-only experiment will be followed by an experiment that includes both audio and video.
- The MM experiment contains two parallel evaluations of video material. One evaluation is by panels of human observers (i.e., subjective testing). The other is by computational models of video quality (i.e., objective models). The objective models are meant to predict the subjective judgments.
- The MM experiment addresses three video resolutions (QCIF, CIF, and VGA) and three types of objective models: full reference (FR), reduced reference (RR), and no reference (NR). FR models have full access to the source video; RR models have limited bandwidth access to the source video; and NR models do not have access to the source video.
- Forty-one subjective experiments provided data to validate objective video quality models. The experiments are divided between the three video resolutions and two frame rates (25fps and 30fps). A common set of carefully chosen video sequences are inserted identically into each experiment at a given resolution, to anchor the video experiments to one another and to assist in comparisons between the subjective experiments. The subjective experiments include processed video sequences with a wide range of quality, and both compression and transmission errors were present in the test conditions. These 41 subjective experiments include 346 source video sequences and 5320 processed video sequences. A total of 984 viewers were involved in the subjective experiments.
- A total of 15 organizations participated in the subjective testing. These organizations are: Acreo, CRC, France Telecom, FUB, IRCCyN, KDDI, Nortel, NTIA, NTT, OPTICOM, Psytechnics, SwissQual, Symmetricom, Verizon, and Yonsei University. Objective models were submitted prior to scene selection, PVS generation, and subjective testing, to ensure that none of the models could be trained on the test material. Of the 31 models that were submitted, 6 were withdrawn, and thus results for 25 are presented in this report. A model is considered in this context to be a model type (i.e., FR, RR, or NR) for a specified resolution (i.e., QCIF, CIF, or VGA).
- Each model is associated with only one video resolution (QCIF, CIF, or VGA). While a proponent often submitted the same type of model for all three video resolutions, these are considered three separate models in the MM Test Plan.

The subjective data were collected using Absolute Category Rating (ACR) with Hidden Reference (ACR-HR). The ACR scale shown to the subjects contains a 5-point scale: excellent, good, fair, poor, and bad. These words are mapped to the numbers 5, 4, 3, 2, and 1 respectively, resulting in Mean Opinion Scores (MOS) ranging from 5 (excellent) to 1 (bad). The ACR-HR scale is on the same 5-point scale; however, post-processing removes the impact of the reference (or original) video sequence from each clip. This results in Differential Mean Opinion Scores (DMOS) ranging from 5 (excellent) to 1 (bad).

The common set video sequences were carefully chosen to span a wide range of content. The following are some of the criteria that were used to select the six common original video clips for each resolution:

- Quality “good” or better (judged by an expert viewer prior to subjective testing).
- Wide range of content type (e.g., video conferencing, news, sports, advertisement, animation, movie, home video).
- Some scenes with high coding complexity and some scenes with low coding complexity.
- At least one scene with high spatial detail and at least one scene with low spatial detail.
- At least one scene with very fast motion (e.g., an object moves across the screen in less than one second).
- Approximately half of the scenes have scene cuts, and approximately half of the scenes do not have scene cuts.
- At least one scene with sharp edges and at least one scene with soft edges.
- Approximately one dimly lit or night scene.

The common PVSs for each resolution were also chosen carefully. These clips were chosen to evenly span the entire range of video quality represented in the MM testing. Also, the common set PVSs contained clips from multiple coding algorithms (e.g., H.264, WM9): some with coding-only impairments, and some with transmission errors at different severities. Including the six originals, there were 30 common clips at each image resolution (QCIF, CIF, VGA).

In addition to the common set, each of the experiments contained eight additional original source video sequences that were also carefully selected using the aforementioned criteria. These 8 sources were sent through 16 different Hypothetical Reference Circuits (HRCs), which included a video encoder (operating at some bit rate), a transmission channel, and a decoder. However, the 16 HRCs were chosen by the individual experiment designer, who had a fair amount of leeway in choosing HRCs. The HRCs were supposed to span approximately the same range of quality as given by a set of example video sequences (i.e., video sequences selected by VQEG to indicate the best and worst quality of interest). The choice of coding algorithm, bit-rate, frame-rate, and transmission errors was left up to the experiment designer, within constraints specified by the MM test plan. These constraints allowed for experimenters to design very different experiments (e.g., one experiment may include a wide variety of coding algorithms without any transmission errors, while another experiment may include one type of coding algorithm only but many cases of transmission errors).

3 OVERVIEW OF APPROACH

The VQEG MM Phase I final report's primary analysis [1] provides confidence intervals (CIs) and significance tests that can be used to determine whether models are significantly different on a per experiment basis. The problem is that there are 13 or 14 individual experiments at each image resolution (QCIF, CIF, and VGA), where each experiment has a different mix of source scenes, codecs, transmission errors, and other quality testing characteristics. Thus, each experiment yields a unique result for relative and absolute performance of the various models. Averaging the primary analysis statistics reduces the amount of data presented, but makes statistical significance testing difficult to compute. Summing the number of times a model is in the group of top-performing models has other problems and issues: significance can be computed but the relative accuracy is lost.

There have also been objections raised that results might be distorted by including the common set of video clips in the analysis of each individual experiment. This is a valid argument since the common set comprises approximately 16-18% of the data in each experiment (24 out of 152 for DMOS, and 30 out of 166 for MOS). Thus, if an objective model by chance were to do especially poorly on the common set, it would be over-penalized. Conversely, if an objective model by chance were to do especially well on the common set, it would be under-penalized (relatively speaking).

3.1 Combining Through Overlapping Subjective Data Sets

Therefore, we chose to examine the MM data in a different fashion. This approach was motivated by the very high laboratory to laboratory correlations of the subjective scores for the common sets, and the fact that this common set spanned the full range of video quality that was presented in the subjective experiments. We utilized the common set at each resolution to map all the subjective scores for all the experiments at a given resolution onto a single scale. This mapping procedure was performed as follows. First, an overall average value (over all subjective experiments) was computed for each of the common video clips (i.e., the grand mean of the common set, where each data point was the average of 13 x 24 or 14 x 24 viewers). The grand mean of the common set over all laboratories can be viewed as the best estimate of the true MOSs for the common set of video clips. Second, for each data set a linear fit was computed (using the standard least-squares technique) between that data set's common clips and these grand means. Third, these linear fits were used to transform all the subjective mean opinion scores and their associated standard deviations onto a single subjective scale.

Finally, redundant copies of the common set were discarded, so that the common set would only appear once in the final superset of mapped subjective data. The particular copy of the common set that was retained in the superset was the one with the highest Pearson correlation to the overall grand mean. We wanted common set scores based on 24 viewers, just like the rest of the data. In actuality, this "best" common set is nearly identical to the grand mean common set since it has a Pearson correlation of 0.98 or higher with the grand mean.

This procedure resulted in the creation of three subjective data supersets: QCIF, CIF, and VGA. These three data supersets of DMOS results are used to analyze FR, RR, and NR models in this document. This use of DMOS for analyzing NR models contradicts the MM Test Plan (which specifies MOS for NR models). However, our motivation for this change was (1) NR models could be directly compared to FR and RR models, which is a comparison that we wanted to make, (2) we felt that the original videos should never have been included in the analysis for NR performance as these models will never be applied to the original videos (similar arguments were used by VQEG to discard three HRCs from one VGA experiment because they exceeded the maximum 4 Mbits/sec bandwidth specification given in the MM test plan), and finally (3) the DMOS and MOS scores are very highly correlated to each other, such that this change has minimal impact on estimating model performance.

3.2 Model Fits & Analysis Metrics

Each objective model was fit to the combined subjective superset by performing a 3rd order monotonic polynomial fit. This fit was done exactly once (i.e., all statistics in this document use the same 3rd order monotonic polynomial fit for each model). Then, the performance metrics in the test plan were computed, including their CIs. Finally, statistical significances between models were computed using RMSE and an F-test, as specified in the VQEG MM Phase I final report. Because of the increased degrees of freedom (i.e., more video clips used simultaneously in the analysis), the F-test on this combined superset of subjective data is better able to differentiate between models than the primary analysis' F-test as applied to an individual experiment. See Sections B.4, B.7, and B.8 of Appendix B for MATLAB code implementing these calculations.

We chose to report significance testing based on only one metric, because this produces a simpler interpretation of results for the reader. RMSE was chosen for statistical significance for the following reasons: (1) the monotonic polynomial 3rd order fit minimizes RMSE, (2) RMSE and Pearson correlation are very closely related, and (3) RMSE tends to have the greatest discrimination capability (i.e., RMSE can better identify differences between models). [2]

3.3 Understanding Resolving Power

In addition to computing the statistics from the MM primary analysis, we also computed the 95%, 90%, 75%, and 68% resolving powers for each model. Resolving power is a statistical technique that enables a user to determine the significance of a quality difference as output by a particular model [2] [3] [4].[3] For example, if a video clip from one video system receives a model output of 2.5 while another video system receives a model output of 4.0 (for a model output difference of 4 – 2.5 = 1.5), and the 95% resolving power of the model is 1.4, then this difference in quality is significant at the 95% level (since 1.5 exceeds 1.4). The availability of four resolving powers allows users to select the confidence appropriate for their application (e.g.,

[2] The results reported in "Comparison of Metrics VQEG MM Data," June 2008, by G. W. Cermak to the VQEG MM project, show that (1) correlation, RMSE, and outlier ratio all measure essentially the same thing, (2) RMSE is better at discriminating between models, and (3) the advantage of RMSE over correlation increases as the number of video samples decreases, and vice versa. These conclusions were also true for the VQEG FR-TV Phase 2 data.

[3] The journal article [4] presents an overview of the resolving power statistic and may be the easiest of the three references to understand.

does their application require 95% confidence that one clip is better than another, or would 75% confidence suffice?). For MATLAB code that computes resolving power, see Section B.5 of Appendix B, [2], or [3].

Resolving power was calculated for each model using the subjective superset associated with that model, after the model was fitted to the superset (see Section 3.2). Thus, the 3rd order polynomial fits published in this document must first be applied to the model output before using the published resolving powers. The resolving power values in this report are thus reported on the [5, 1] ACR scale used by the VQEG MM Phase I experiment. On this scale, 5 represents excellent quality and 1 represents bad quality, so decreasing scores indicate a drop in quality.

The following provides a more detailed description of resolving power. Resolving power is defined mathematically as the delta Video Metric (VM) value above which the conditional subjective-score distributions have means that are statistically different from each other at a given confidence level (e.g., 95% significance level). Put more simply, 95% resolving power is a delta VM value that acts like a 95% confidence interval. When two video sequences' VM differ by more than this delta value, we have 95% confidence that a subjective test would also indicate that one video sequence has significantly different quality than the other. When two video sequences' VMs differ by less than this delta value, the objective model cannot tell the difference between the two video clips' quality (i.e., we have less than 95% confidence that a subjective test would agree with the VM results).

Suppose we have two PVSs; PVS A (PVS_A) with video metric value VM_A and PVS B (PVS_B) with video metric value VM_B, such that

$$VM_A \geq VM_B.$$

If

$$(VM_A - VM_B) \geq 95\% \text{ resolving power,}$$

then we can be 95% confident that a subjective test would find that PVS_A has higher quality than PVS_B. Conversely, there is a 5% chance that the objective model has made a mistake (i.e., PVS_A and PVS_B have the same subjective quality, or PVS_A has lower quality than PVS_B). Resolving power takes into account the uncertainty in the subjective data. If

$$(VM_A - VM_B) < 95\% \text{ resolving power,}$$

then the objective model cannot distinguish between the quality of PVS_A and the quality of PVS_B at the 95% confidence level.

95% resolving power yields a single number for each MM objective model at each resolution (VGA, CIF, and QCIF). This gives the user an easy way to understand the model's accuracy and limitations. End-users will realize that VM differences less than the 95% resolving power mean that those clips' video qualities cannot be distinguished as being different by the VM.

A word of caution is in order. Resolving power should not be used to directly compare the performance of two different objective models. The reason is that the mapped outputs from different models may span different portions of the subjective scale.

3.4 Comparing Different Video Resolutions

Several characteristics of the MM Test Plan prevent models at different resolutions from being directly compared. The viewing angle between pixels is different for each resolution, as is the angle extended by the video picture (encompassed by the entire image). The distribution of HRCs is quite different from one resolution to another (i.e., the frequency of each coding algorithm and transmission errors are dissimilar). Thus, the methods presented herein cannot be used to join QCIF, CIF, and VGA results from the VQEG MM Phase I Test into a single comparison.

3.5 Model Identification & PSNR Reference Model

Throughout this report, each model is identified by a randomly assigned letter, the type of model (FR, RR, or NR), and the video resolution (QCIF, CIF, and VGA).

PSNR is included as a reference metric for every analysis. PSNR is a FR model that utilizes one constant delay for each video sequence (see Appendix A). The MATLAB code used to compute PSNR is given in Section B.1 of Appendix B.

Because PSNR is widely used for estimating video quality, PSNR's performance can be used as a benchmark for judging the performance of a model. For this report, the statistical significance tests that compare a model's performance with PSNR will be dependent upon the model type. We will determine if FR models perform statistically better than PSNR (i.e., otherwise, PSNR could be used instead since this is also an FR model). On the other hand, we will determine if RR and NR models perform statistically equivalently to or better than PSNR, since RR and NR models operate in an environment where PSNR is not available. While objections can be raised concerning the use of PSNR as a minimum performance benchmark and our interpretation of this minimim benchmark, no better benchmark has yet been proposed.

4 MM COMMON SET ANALYSIS

Table 1, Table 3, and Table 5 contain for each data superset (QCIF, CIF, and VGA) the optimal linear fits required to transform that experiment's common set subjective scores to the grand mean of the common set. These fits are also applied to scale all the subjective DMOSs and their associated standard deviations to create the combined data supersets. For MATLAB code to compute and apply these fits, see Section B.3 of Appendix B. Table 2, Table 4, and Table 6 contain the Pearson correlation between the common sets for all the experiments. These calculations use only the DMOSs of the twenty-four common set PVSs (original sequences are not included). These tables also list the Pearson correlation between each experiment's common set and the grand mean (GM), which is highlighted in yellow. Within the yellow highlighted row, the experiment whose common set was selected for retention in the larger superset is shown in bold and underlined. Figure 1, Figure 2, and Figure 3 contain a scatter plot between the common set subjective scores (after the fit) and the grand mean for each individual experiment. The confidence interval of the fitted DMOS extends horizontally in turquoise. Each data point is plotted as a square, so that overlapping data points can be distinguished.

These tables and figures show the high repeatability of the common set scores from laboratory to laboratory, from experiment to experiment, and from country to country. Note that these common set sequences were contained within larger experiments, where the rest of the video clips differed, but the testing methodology remained the same (e.g., the same subjective testing procedure was used). These data demonstrate a high degree of repeatability for well conducted subjective testing.

4.1 QCIF Mapping Results

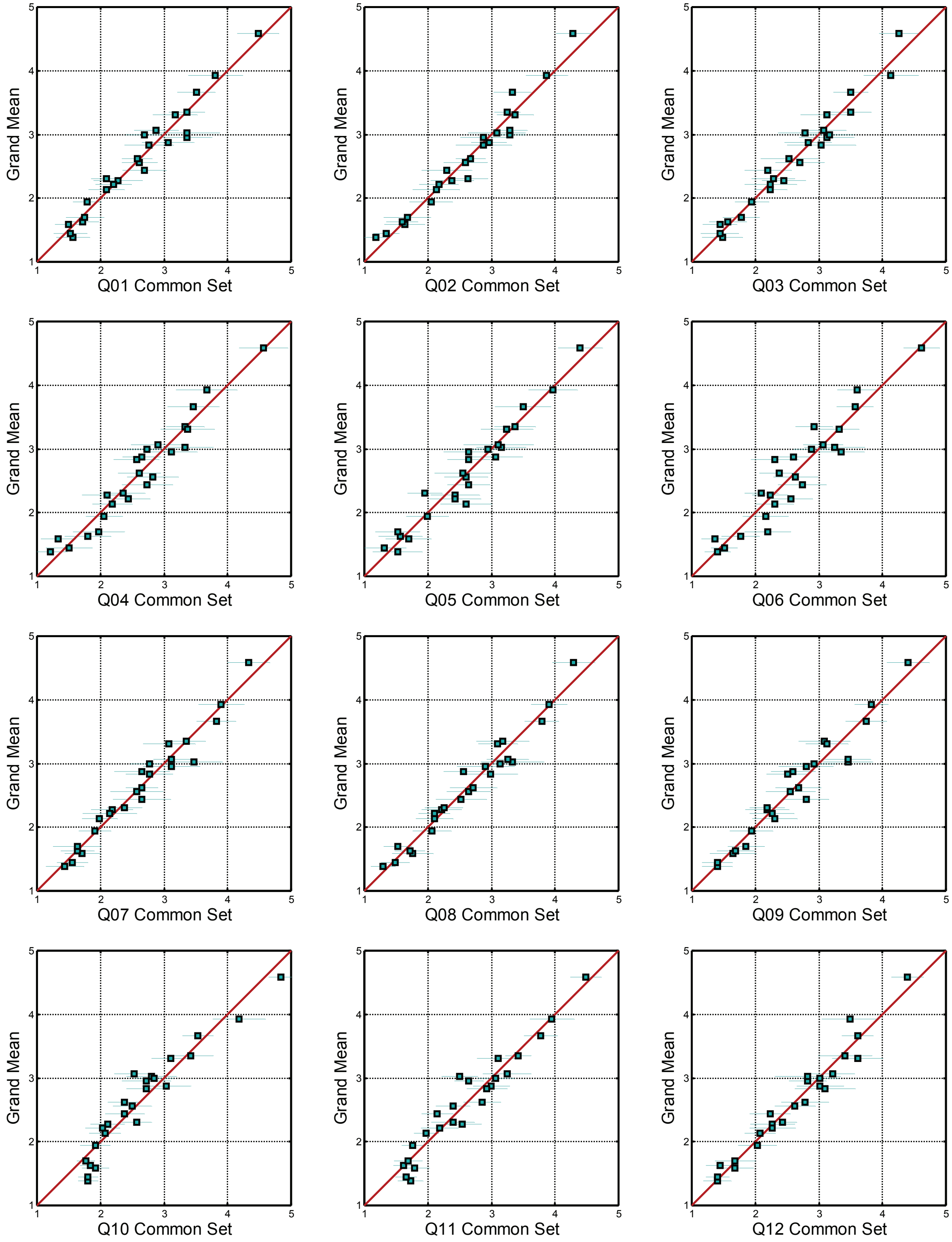
Table 1. Gain and Offset Required to Transform QCIF Experiments

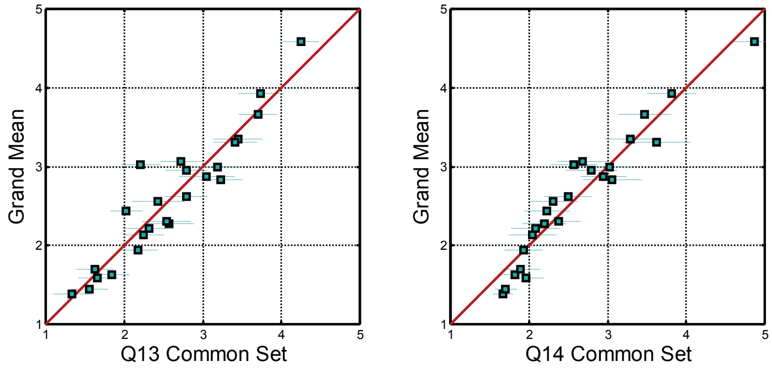
Experiment	Gain	Offset
Q01	0.898908	0.215169
Q02	0.996353	-0.154169
Q03	1.011425	-0.080195
Q04	1.020663	-0.156217
Q05	1.031032	-0.325797
Q06	0.867632	0.244508
Q07	0.939485	0.026831
Q08	0.924049	0.173665
Q09	0.988002	-0.041746
Q10	0.922892	0.686236
Q11	0.850353	0.516155
Q12	0.946564	0.138782
Q13	0.875502	0.200190
Q14	0.906594	0.641635

Table 2. QCIF Common Set Pearson Correlations

	Q01	Q02	Q03	Q04	Q05	Q06	Q07	Q08	Q09	Q10	Q11	Q12	Q13	Q14
Q01	1.00													
Q02	0.94	1.00												
Q03	0.95	0.97	1.00											
Q04	0.97	0.94	0.94	1.00										
Q05	0.96	0.95	0.95	0.94	1.00									
Q06	0.95	0.91	0.91	0.98	0.92	1.00								
Q07	0.98	0.95	0.95	0.97	0.95	0.94	1.00							
Q08	0.95	0.97	0.96	0.95	0.95	0.92	0.99	1.00						
Q09	0.95	0.95	0.93	0.96	0.96	0.96	0.98	0.98	1.00					
Q10	0.93	0.92	0.93	0.91	0.92	0.88	0.93	0.91	0.90	1.00				
Q11	0.92	0.95	0.96	0.89	0.93	0.86	0.93	0.94	0.91	0.94	1.00			
Q12	0.94	0.98	0.96	0.94	0.95	0.90	0.94	0.96	0.93	0.92	0.96	1.00		
Q13	0.88	0.94	0.95	0.87	0.90	0.83	0.88	0.91	0.85	0.91	0.96	0.96	1.00	
Q14	0.92	0.93	0.94	0.91	0.91	0.88	0.91	0.91	0.89	0.97	0.95	0.95	0.95	1.00
QGM	0.98	0.98	**0.98**	0.97	0.97	0.95	0.98	0.98	0.97	0.96	0.97	0.98	0.94	0.96

Figure 1. QCIF: scatter plot of each experiment's fitted common set to the grand mean.





4.2 CIF Mapping Results

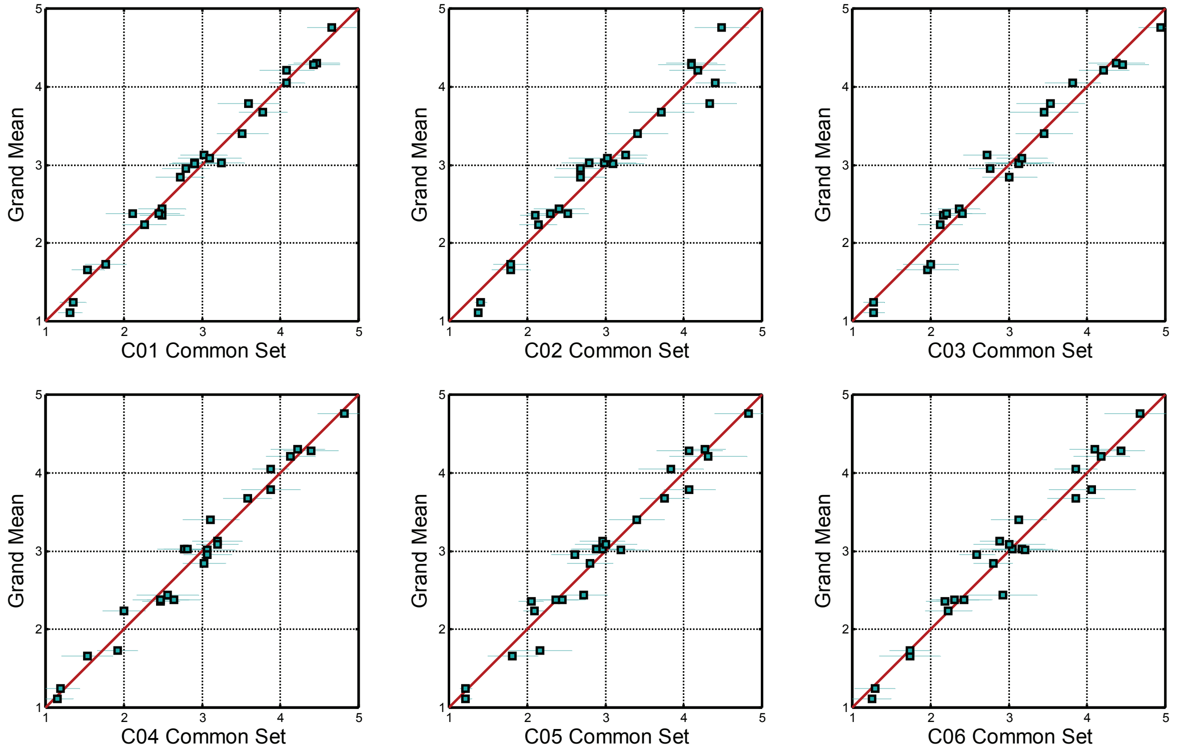
Table 3. Gain and Offset Required to Transform CIF Experiments

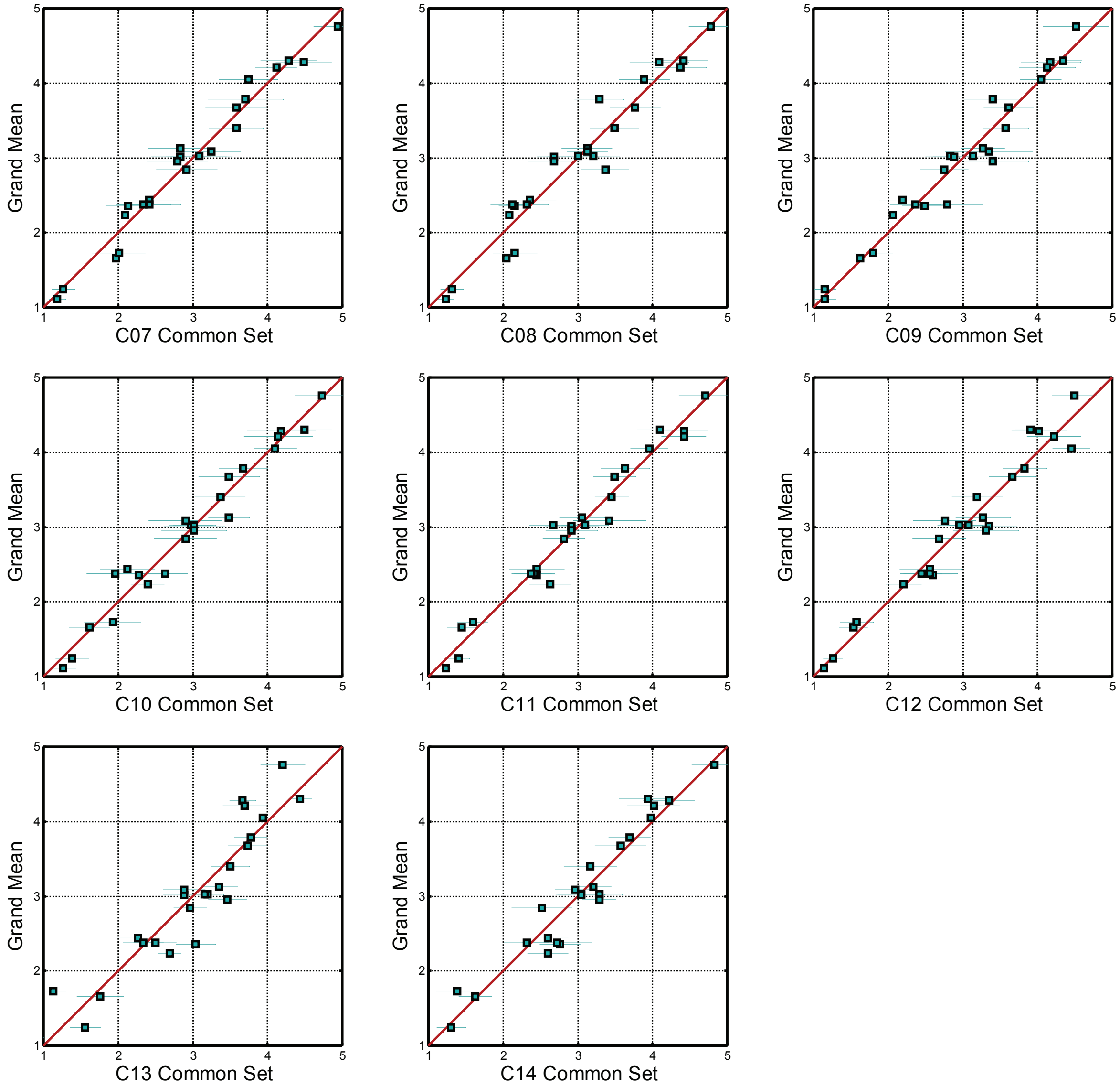
Experiment	Gain	Offset
C01	0.912755	0.320696
C02	0.924684	0.441912
C03	0.968177	0.183927
C04	1.024623	-0.262583
C05	0.954368	0.218562
C06	0.981117	-0.023520
C07	0.993841	0.101694
C08	0.968109	0.180760
C09	1.036942	-0.015097
C10	0.934882	0.094390
C11	0.860924	0.333554
C12	0.948036	0.190669
C13	0.935441	-0.041275
C14	0.973564	0.002829

Table 4. CIF Common Set Pearson Correlations

	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12	C13	C14
C01	1.00													
C02	0.97	1.00												
C03	0.97	0.95	1.00											
C04	0.97	0.96	0.97	1.00										
C05	0.97	0.98	0.98	0.97	1.00									
C06	0.97	0.97	0.97	0.97	0.99	1.00								
C07	0.98	0.96	0.99	0.97	0.98	0.97	1.00							
C08	0.97	0.94	0.98	0.95	0.96	0.94	0.98	1.00						
C09	0.96	0.94	0.96	0.97	0.94	0.93	0.96	0.95	1.00					
C10	0.98	0.97	0.96	0.97	0.96	0.95	0.96	0.96	0.96	1.00				
C11	0.97	0.95	0.96	0.97	0.95	0.96	0.97	0.94	0.97	0.96	1.00			
C12	0.96	0.96	0.93	0.97	0.95	0.96	0.93	0.91	0.96	0.96	0.96	1.00		
C13	0.93	0.90	0.88	0.91	0.88	0.89	0.89	0.89	0.94	0.93	0.92	0.94	1.00	
C14	0.96	0.93	0.94	0.96	0.93	0.95	0.94	0.91	0.96	0.95	0.97	0.98	0.95	1.00
CGM	**0.99**	0.98	0.98	0.99	0.98	0.98	0.99	0.97	0.98	0.98	0.98	0.98	0.94	0.97

Figure 2. CIF: scatter plot of each experiment's fitted common set to the grand mean.




Grand Mean
C07 Common Set
C08 Common Set
C09 Common Set
C10 Common Set
C11 Common Set
C12 Common Set
C13 Common Set
C14 Common Set
1
2
3
4
5

4.3 VGA Mapping Results

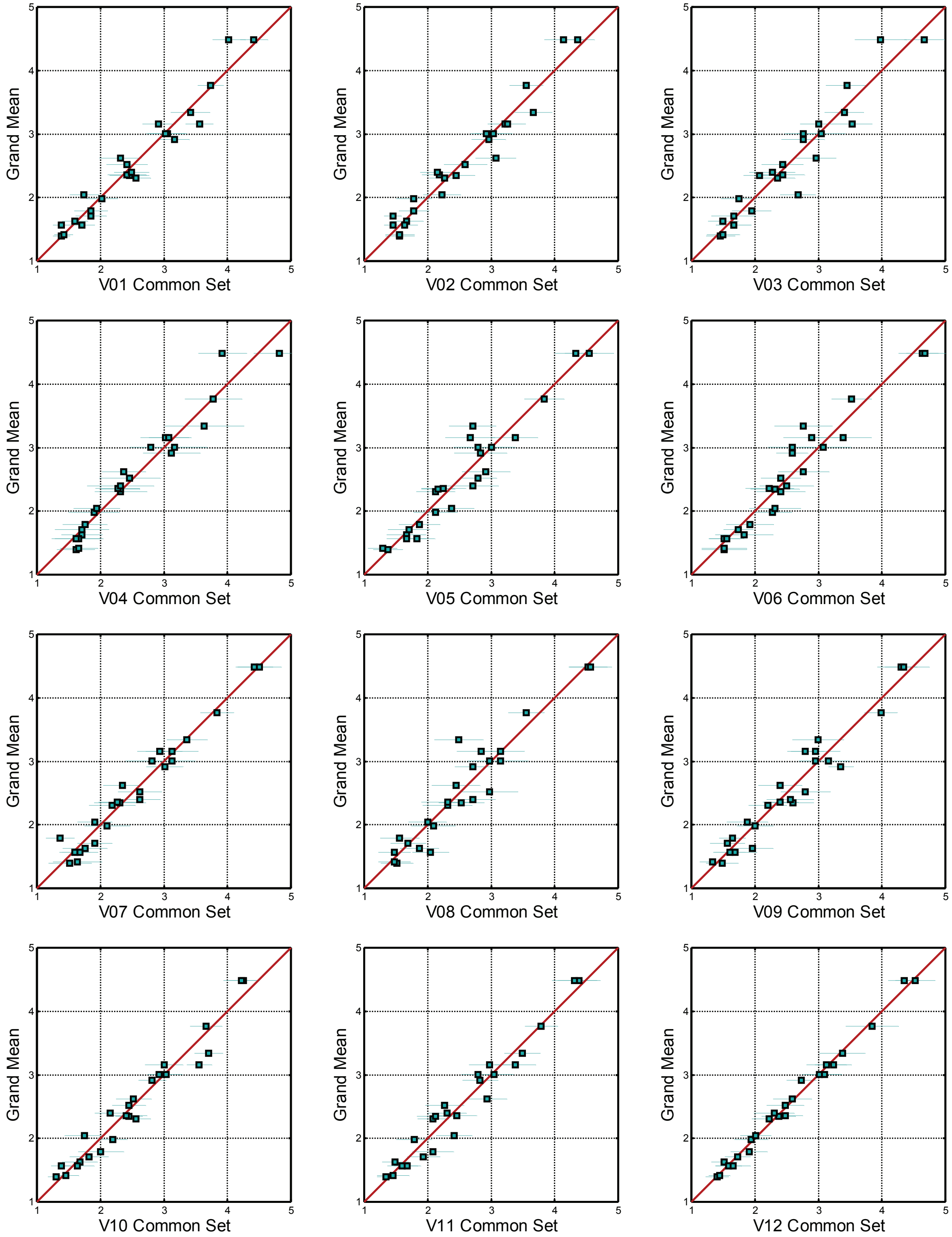
Table 5. Gain and Offset Required to Transform VGA Experiments

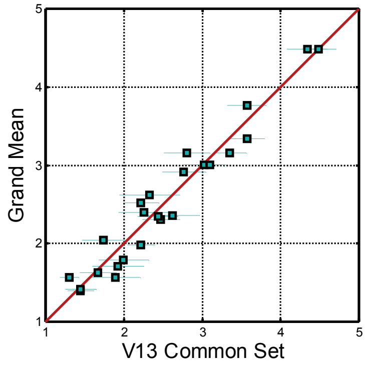
Experiment	Gain	Offset
V01	0.856987	0.382604
V02	0.886580	0.412824
V03	0.976888	0.072498
V04	1.132407	-0.182461
V05	1.004334	-0.012478
V06	1.072270	-0.274384
V07	0.944088	0.020895
V08	1.062632	-0.081221
V09	0.953184	-0.024875
V10	0.886926	0.267253
V11	0.891773	0.298192
V12	0.863358	0.321610
V13	0.881207	0.304846

Table 6. VGA Common Set Pearson Correlations

	V01	**V02**	**V03**	**V04**	**V05**	**V06**	**V07**	**V08**	**V09**	**V10**	**V11**	**V12**	**V13**
V01	1.00												
V02	0.94	1.00											
V03	0.90	0.97	1.00										
V04	0.97	0.95	0.91	1.00									
V05	0.90	0.93	0.95	0.91	1.00								
V06	0.91	0.93	0.94	0.92	0.97	1.00							
V07	0.97	0.95	0.92	0.96	0.94	0.95	1.00						
V08	0.91	0.91	0.89	0.91	0.97	0.96	0.95	1.00					
V09	0.96	0.94	0.90	0.95	0.94	0.92	0.97	0.96	1.00				
V10	0.99	0.96	0.92	0.96	0.89	0.92	0.95	0.89	0.93	1.00			
V11	0.93	0.97	0.98	0.95	0.95	0.94	0.94	0.90	0.92	0.94	1.00		
V12	0.97	0.97	0.95	0.97	0.95	0.96	0.97	0.94	0.96	0.98	0.98	1.00	
V13	0.98	0.93	0.91	0.95	0.89	0.92	0.96	0.89	0.93	0.99	0.94	0.97	1.00
VGM	0.98	0.98	0.96	0.98	0.96	0.97	0.98	0.95	0.97	0.98	0.98	**1.00**	0.97

Figure 3. VGA: scatter plot of each experiment's common set to the grand mean.




Grand Mean
V13 Common Set
1
2
3
4
5
1
2
3
4
5

5 SUPERSET ANALYSIS, OBJECTIVE FITS, AND RESOLVING POWER

This section calculates the values for the three performance metrics specified in the VQEG MM Phase I final report [1] (Pearson correlation, RMSE, and outlier ratio), except that the calculations are performed on the subjective data supersets that result from the mappings performed in Section 4. The analysis presented in this section is computed on a per-clip basis. In the tables below, the column marked "lower CI" indicates the worst value in the 95% confidence interval (i.e., worst expected performance); while the column marked "upper CI" indicates the best value in the 95% confidence interval (i.e., best expected performance). For MATLAB code to compute Pearson correlation, RMSE, outlier ratio, and the respective confidence intervals, see Sections B.7 and B.8 of Appendix B.

This section also presents group rankings, which present pair wise statistical comparisons between all models. These group rankings indicate whether each model's performance was statistically better, equivalent, or worse than the performance of the other models. For reasons explained in Section 3.2, the group rankings for the models are computed using only the RMSE F-test results. These RMSE rank groupings are computed as follows. First, the models were sorted from best performance (lowest RMSE) to worst performance (highest RMSE). The best model is then selected and all other models that are statistically equivalent at the 95% significance level (using the F-test) to the best are identified as belonging to Group 1 (G1). The process is repeated for the next best model to produce Group 2 (G2) and so on. This yields performance groupings. Redundant groupings are then merged and renumbered. MATLAB code to compute this F-test is included in Section B.9 of Appendix B.

All "Xs" in a column indicate that those models are statistically equivalent to the model marked with an asterisk ("X*"). To find out which other models are statistically equivalent to a particular model, follow that model's row to the right until you reach the only column with an asterisk in that row. All models in that column with an "X" or an "X*" are statistically equivalent to the model under consideration. When rank groupings are calculated in this manner, models can belong to multiple groups simultaneously. If two models in a column both have an asterisk, this indicates that those models yield the same statistical equivalence results.

The group rankings presented in this section are influenced by the distribution of HRCs among coding algorithms, and the distribution of HRCs between compression artifacts only and compression artifacts compounded with transmission errors. For example, each superset contains three to four times more H.264 HRCs than Real Video 10 HRCs. See Section 5 for the distribution of HRCs among these variables for each superset.

5.1 QCIF Results

Table 7, Table 8, and Table 9 contain QCIF rankings for Pearson correlation, RMSE, and outlier ratio. In these tables we use "Lower CI" and "Upper CI" for those values which reflect the worst and best expected performance of the model, respectively.

Table 7. QCIF: Pearson Correlation and its CI

FR Models	**Lower CI**	**Correlation**	**Upper CI**
PSNR	0.674	0.698	0.721
A	0.829	0.843	0.856
B	0.783	0.800	0.816
C	0.795	0.811	0.826
D	0.812	0.827	0.841
RR Models	**Lower CI**	**Correlation**	**Upper CI**
E	0.787	0.804	0.820
F	0.816	0.831	0.845
NR Models	**Lower CI**	**Correlation**	**Upper CI**
G	0.674	0.698	0.721
H	0.630	0.657	0.683

Table 8. QCIF: RMSE and its CI, and Group Rankings

FR Models	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
PSNR	0.707	0.684	0.662					X*	
A	0.531	0.514	0.498	X*	X				
B	0.593	0.573	0.555				X*		
C	0.578	0.559	0.541				X*		
D	0.556	0.538	0.521		X	X*			
RR Models	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
E	0.587	0.568	0.550				X*		
F	0.549	0.531	0.515	X	X*	X			
NR Models	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
G	0.707	0.684	0.663					X*	
H	0.745	0.720	0.698						X*

Table 9. QCIF: Outlier Ratio and its CI

FR Models	Lower CI	Outlier Ratio	Upper CI
PSNR	0.664	0.642	0.620
A	0.503	0.480	0.457
B	0.556	0.533	0.510
C	0.551	0.528	0.505
D	0.498	0.475	0.452
RR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
E	0.578	0.555	0.532
F	0.550	0.528	0.505
NR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
G	0.639	0.617	0.594
H	0.668	0.646	0.624

Table 10 contains the resolving power (RP) for each QCIF model, computed at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power.

Table 11 contains the 3rd order monotonic polynomial fit for each model to the QCIF superset's ACR scale. The fits in Table 11 remove any non-linearity between the model and subjective scores, and presents model results on the [5, 1] ACR scale. Our presumption is that the model developers want to remove this non-linearity from their model. It should be noted that after the polynomial mapping, not all models span the entire ACR [5, 1] scale. This should be considered when examining resolving power values. The fits shown in the table are utilized for all the data analyses in this report. MATLAB code to compute these fits is given in Section B.4 of Appendix B. The fitted model values (VM_{fit}) is computed from the raw model values (VM) as follows:

$$VM_{fit} = A3 * VM^3 + A2 * VM^2 + A1 * VM + A0$$

Table 10. QCIF: Resolving Power

FR Models	95% RP	90% RP	75% RP	68% RP
PSNR	1.56	1.26	0.70	0.49
A	1.33	1.01	0.50	0.34
B	1.49	1.11	0.54	0.37
C	1.41	1.08	0.55	0.38
D	1.40	1.04	0.53	0.36
RR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
E	1.41	1.09	0.58	0.40
F	1.33	1.03	0.54	0.37
NR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
G	1.63	1.30	0.68	0.47
H	1.69	1.35	0.72	0.49

Table 11. QCIF: Objective Model Fits

FR Models	**A3**	**A2**	**A1**	**A0**
PSNR	-0.0000554377526622	0.0035819545229566	0.0447411137535195	0.5327704942730040
A	-0.0645584143596585	0.6587017237209710	-1.1237282043629700	2.0109176496909300
B	-0.0000227618453518	0.0006063755601715	0.0636684505062855	2.5464010879383800
C	-0.0478299421460788	0.3944209388933350	0.0177852421212146	0.8136716466302330
D	-0.0592497130563754	0.5308335576199670	-0.6073444105962830	1.6856374933596900
RR Models	**A3**	**A2**	**A1**	**A0**
E	-0.0000189476476094	-0.0007283419425248	0.1936903493867400	-0.7910221498111830
F	-0.0000495876837159	0.0025765736871902	0.0831727046606984	0.4562553430767220
NR Models	**A3**	**A2**	**A1**	**A0**
G	-0.0706062299310762	0.6019025833409500	-0.7235585885787910	1.7442210192509400
H	0.0531905898415887	-0.2809949536214140	1.0117825737663500	1.3124783202729900

Our interpretation of the QCIF results is as follows:

- The top group (column G1 on Table 8) contains two models: FR model A and RR model F. All other models are statistically worse than model A.

- RR model F is statistically equivalent to both FR model A and FR model D (G2). FR model D is statistically worse than FR model A, statistically equivalent to RR model F, and statistically better than all other models (column G3 on Table 8).

- All FR and RR models are statistically better than PSNR (see Table 8).

- NR model G is statistically equivalent to PSNR, and NR model H is statistically worse than PSNR (columns G5 and G6 on Table 8).

- The resolving power values in Table 10 (applied after the fits from Table 11) allow users to compare their model scores from video sequences, and understand the accuracy of that comparison.

5.2 CIF Results

Table 12, Table 13, and Table 14 contain CIF rankings for Pearson correlation, RMSE, and outlier ratio.

Table 12. CIF: Pearson Correlation and its CI

FR Models	Lower CI	Correlation	Upper CI
PSNR	0.614	0.642	0.668
I	0.776	0.794	0.810
J	0.738	0.759	0.777
K	0.834	0.847	0.860
L	0.777	0.795	0.811
RR Models	**Lower CI**	**Correlation**	**Upper CI**
M	0.754	0.773	0.791
N	0.754	0.773	0.791
NR Models	**Lower CI**	**Correlation**	**Upper CI**
O	0.442	0.478	0.513
P	0.481	0.516	0.549

Table 13. CIF: RMSE and its CI, and Group Rankings

FR Models	Lower CI	RMSE	Upper CI	G1	G2	G3	G4	G5
PSNR	0.759	0.735	0.711				X*	
I	0.602	0.582	0.564		X*			
J	0.645	0.624	0.604			X*		
K	0.526	0.509	0.493	X*				
L	0.601	0.582	0.563		X*			
RR Models	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.628	0.607	0.588			X*		
N	0.628	0.607	0.588			X*		
NR Models	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.870	0.841	0.815					X*
P	0.848	0.821	0.795					X*

Table 14. CIF: Outlier Ratio and its CI

FR Models	Lower CI	Outlier Ratio	Upper CI
PSNR	0.692	0.671	0.649
I	0.562	0.539	0.516
J	0.589	0.567	0.544
K	0.530	0.507	0.484
L	0.572	0.550	0.527
RR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
M	0.592	0.569	0.546
N	0.588	0.566	0.543
NR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
O	0.719	0.698	0.677
P	0.709	0.688	0.666

Table 15 contains the resolving power for each CIF model, computed at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power.

Table 16 contains the 3rd order monotonic polynomial fit for each model to the CIF superset's ACR scale. The fits in Table 16 remove any non-linearity between the model and subjective scores, and present model results on the [5, 1] ACR scale. Our assumption is that the model developers want to remove this non-linearity from their model. It should be noted that after the polynomial mapping, not all models span the entire ACR [5, 1] scale. This should be considered when examining resolving power values. The fits shown in the table are utilized for all the data analyses in this report.

Table 15. CIF: Resolving Power

FR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
PSNR	1.67	1.34	0.75	0.52
I	1.48	1.11	0.57	0.39
J	1.57	1.21	0.60	0.40
K	1.29	1.00	0.53	0.37
L	1.51	1.14	0.57	0.39
RR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
M	1.53	1.15	0.58	0.40
N	1.53	1.15	0.58	0.40
NR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
O	1.65	1.43	0.85	0.56
P	1.76	1.41	0.81	0.57

Table 16. CIF: Objective Model Fits

FR Models	**A3**	**A2**	**A1**	**A0**
PSNR	0.0000155249913691	-0.0020474466695046	0.1761688292755750	-0.3314058293091570
I	-0.0000215099724017	0.0005023205894147	0.0674220640835921	2.7717322020514600
J	-0.0554096261962689	0.5158273233297840	-0.6127261648408010	1.7652992312409000
K	-0.0528355557100217	0.4355953487353320	-0.2239447745810950	1.3525220981352600
L	0.0311032671067133	-0.3409223026856490	2.1195433207182700	-0.9279227376397380
RR Models	**A3**	**A2**	**A1**	**A0**
M	-0.0000549175426936	0.0037639478316817	0.0354867870336490	1.1104050517530900
N	-0.0000533111148658	0.0035877515809232	0.0410582034010182	1.0512902990867500
NR Models	**A3**	**A2**	**A1**	**A0**
O	0.0612817755026727	-0.3654238625418480	1.0761136588766900	1.7217361306421200
P	-0.0405221841279052	0.2901862032213430	0.1373017773794580	1.2815608635921400

Our interpretation of the CIF results is as follows:

- The top group (column G1 in Table 13) contains one model: FR model K. All other models are statistically worse than model K.

- FR models L and I are statistically equivalent to each other and statistically better than all remaining models (that is, ignoring model K; see column G2 in Table 13).

- The remaining FR models and all RR models are statistically better than PSNR (see Table 13).

- The two NR models are statistically worse than PSNR (column G5 in Table 13).

- The resolving power values in Table 15 (applied after the fits from Table 16) allow users to compare their model scores from video sequences, and understand the accuracy of that comparison.

5.3 VGA Results

Table 17, Table 18, and Table 19 contain VGA rankings for Pearson correlation, RMSE, and outlier ratio.

Table 17. VGA: Pearson Correlation and its CI

FR Models	**Lower CI**	**Correlation**	**Upper CI**
PSNR	0.704	0.727	0.749
Q	0.802	0.818	0.834
R	0.718	0.741	0.761
S	0.785	0.803	0.820
T	0.779	0.797	0.814
RR Models	**Lower CI**	**Correlation**	**Upper CI**
U	0.781	0.799	0.816
V	0.782	0.800	0.816
W	0.782	0.800	0.817
NR Models	**Lower CI**	**Correlation**	**Upper CI**
X	0.367	0.408	0.447
Y	0.389	0.429	0.468

Table 18. VGA: RMSE and its CI, and Group Rankings

FR Model	Lower CI	RMSE	Upper CI	G1	G2	G3	G4	G5
PSNR	0.725	0.701	0.678				X*	
Q	0.607	0.586	0.567	X*	X			
R	0.710	0.686	0.663				X*	
S	0.629	0.608	0.588	X	X*	X		
T	0.638	0.617	0.596		X	X*		
RR Model	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**
U	0.635	0.613	0.593		X	X*		
V	0.634	0.613	0.593		X	X*		
W	0.634	0.612	0.592		X	X*		
NR Model	**Lower CI**	**RMSE**	**Upper CI**	**G1**	**G2**	**G3**	**G4**	**G5**
X	0.965	0.932	0.901					X*
Y	0.954	0.922	0.891					X*

Table 19. VGA: Outlier Ratio and its CI

FR Models	Lower CI	Outlier Ratio	Upper CI
PSNR	0.661	0.638	0.615
Q	0.580	0.556	0.533
R	0.648	0.624	0.601
S	0.582	0.558	0.534
T	0.588	0.564	0.540
RR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
U	0.599	0.575	0.551
V	0.603	0.579	0.556
W	0.601	0.578	0.554
NR Models	**Lower CI**	**Outlier Ratio**	**Upper CI**
X	0.771	0.751	0.730
Y	0.744	0.722	0.701

Table 20 contains the resolving power for each VGA model, computed at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power. Where ">4" is reported, the 95% resolving power is larger than the full model output range after application of the polynomial fit.

Table 21 contains the 3rd order monotonic polynomial fit for each model to the VGA superset's ACR scale. The fits in Table 21 remove any non-linearity between the model and subjective scores, and present model results on the [5, 1] ACR scale. Our assumption is that the model developers want to remove this non-linearity from their model. It should be noted that after the polynomial mapping, not all models span the entire ACR [5, 1] scale. This should be considered when examining resolving power values. The fits shown in the table are utilized for all the data analyses in this report.

Table 20. VGA: Resolving Power

FR Model	95% RP	90% RP	75% RP	68% RP
PSNR	1.66	1.29	0.71	0.50
Q	1.47	1.14	0.58	0.40
R	1.66	1.32	0.68	0.44
S	1.56	1.18	0.58	0.40
T	1.55	1.17	0.61	0.42
RR Model	**95% RP**	**90% RP**	**75% RP**	**68% RP**
U	1.48	1.14	0.61	0.43
V	1.48	1.14	0.61	0.43
W	1.48	1.13	0.60	0.42
NR Model	**95% RP**	**90% RP**	**75% RP**	**68% RP**
X	1.86	1.59	0.94	0.59
Y	> 4	1.83	0.73	0.50

Table 21. VGA: Objective Model Fits

FR Models	A3	A2	A1	A0
PSNR	-0.0001120662433072	0.0104613167946570	-0.2027343064403860	3.2773252926750500
Q	0.0177702921049815	-0.1232869228805290	1.0844868281571900	0.5931698767236280
R	-0.0579971298187437	0.6028801644658410	-1.0317689394754500	2.1675219674857700
S	-0.0476143362795590	0.5103481194151070	-0.8778532299915370	2.4724398750069500
T	-0.0000197618269059	0.0004830899888138	0.0674014255161652	2.9116567979554800
RR Models	**A3**	**A2**	**A1**	**A0**
U	-0.0001168532593714	0.0091390934154342	-0.1044083755629860	2.2334062913107800
V	-0.0001187752582736	0.0093535515890485	-0.1112856705677230	2.2982659245898200
W	-0.0001202966366516	0.0094697512372164	-0.1136401545888130	2.3028876014511900
NR Models	**A3**	**A2**	**A1**	**A0**
X	0.1884907856461040	-1.3632165623179700	3.3975826401220200	0.4775300713254210
Y	0.0173861525531211	-0.3340122930866810	2.0361614893827300	-0.1191128790853110

Our interpretation of the VGA results is as follows:

- The top group (column G1 of Table 18) contains two models: FR models Q and S. All other models are statistically worse than model Q.

- All RR models and all FR models except R are statistically better than PSNR. FR model R is statistically equivalent to PSNR (see Table 18).

- Both NR models are statistically worse than PSNR (columns G4 and G5 of Table 18).

- The resolving power values in Table 20 (applied after the fits from Table 21) allow users to compare their model scores from video sequences, and understand the accuracy of that comparison.

6 MODEL RESPONSE TO IMPAIRMENT TYPE

Two important categories of impairment types exist in the VQEG MM Phase I experiments. The first category is video sequences that contain compression artifacts only, which will be referred to as "coding only." The second category is video sequences that contain coding plus simulated transmission errors or live network errors, which will be referred to as "transmission errors." Unfortunately, there was no overall planning or coordination of experiments to assist in analyzing these impairment sub-categories directly for each experiment.

Use of the subjective data supersets (see Section 3.1) allows us to separate processed video sequences by impairment type, yet retain enough video sequences to reach powerful conclusions. This type of analysis would not be possible with just the individual data sets for three main reasons. The first reason is that the individual data sets utilize a very limited number of scenes and HRC types, so there is simply too little data to compute meaningful comparisons of model performance versus impairment type. The second reason is that the individual experiments were not designed to answer these questions. Thus, when a few HRCs within one experiment are considered in isolation from the rest, the experiment may become unbalanced (e.g., span a small range of quality). There is a third, more subtle, reason that illustrates the power of using the subjective data superset, namely, the quantification of fixed quality biases with respect to an individual experiment that might be present in some of the objective models. These biases would not show up in the analysis of the individual data sets but would show up in the analysis of the data superset. For example, consider two subjective experiments that contain only one coding algorithm each (e.g., H.264 or VC-1). Consider a hypothetical model that has a quality bias, where VC-1 is always under-penalized and H.264 is always over-penalized, by some fixed amount. This bias does not impact the model's accuracy when measured against a subjective experiment that contains only (or predominantly) one coding algorithm. However, when those experiments are combined into a single experiment, the bias becomes readily apparent. Models with these types of biases would be penalized in the superset analysis but would not be penalized in the analysis of the individual experiments. Conversely, models without these types of biases would be rewarded in the superset analysis. Such an experiment bias may result from other factors, such as the quality impact of the source video scenes associated with one experiment being particularly easy or difficult for the model to predict.

For the impairment type analysis that will be presented in this section, only the following statistics will be reported: Pearson correlation, RMSE, outlier ratio, and statistical significance using RMSE. Confidence intervals, statistical significance using Pearson correlation, and statistical significance using outlier ratio are eliminated to simplify the data presentation. These extra numbers resulted in an overly complicated presentation, which can hinder understanding of the data.

6.1 QCIF Results

This section examines QCIF model performance for two major subdivisions of the video clips: video clips with only coding artifacts, and video clips with coding artifacts plus transmission errors. For each major category, a further subdivision is examined to determine how the models perform with respect to codec type. Codec types are divided into the following five sub-categories: H.264, MPEG-4 (excluding H.264), Video Codec 1 (VC-1, also known as Windows Media 9), Real Video 10 (RV-10), and Other.

The "Other" sub-category includes a variety of codecs each used for one to five HRCs: H.263 (5 HRCs), H.261 (2 HRCs), MPEG-1 (2 HRCs), DivX (2 HRCs & 1 common clip), Sorenson (1 HRC), and Cinepak (1 HRC). Some of these "Other" codecs were created using proprietary codecs, and some utilized non-standard variations of the mentioned codecs.

Table 22 shows the number of QCIF video clips associated with each major category, henceforth abbreviated as "Coding Only," and "Transmission Errors." Note that codecs are not evenly balanced (i.e., different number of clips) with respect to the "Coding Only" and "Transmission Errors" categories. Thus, when multiple codecs are combined, this will skew results more heavily toward those systems that have more clips. These imbalances are also present in the global analysis presented in Section 5.1.

Table 22. QCIF: Number of Video Clips in Each Category

CODEC	CODING ONLY	TRANSMISSION ERRORS
H.264	387	199
MPEG-4	360	280
VC-1	117	192
RV-10	113	64
Other	88	16
TOTAL	1065	751

Table 23 lists the following statistics computed using all the video sequences in the QCIF superset: Pearson correlation, RMSE, outlier ratio, and the ranking groups using RMSE. The information in Table 23 is identical to that presented in Table 8, but is reproduced here for easy comparisons. Table 24 repeats this analysis, but shows only those video sequences that contain coding artifacts. Table 25 shows those video sequences that contain transmission errors.

Table 23. QCIF Data: All Video Sequences

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6
PSNR	0.698	0.684	0.642					X*	
A	0.843	0.514	0.480	X*	X				
B	0.800	0.573	0.533				X*		
C	0.811	0.559	0.528				X*		
D	0.827	0.538	0.475		X	X*			
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
E	0.804	0.568	0.555				X*		
F	0.831	0.531	0.528	X	X*	X			
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
G	0.698	0.684	0.617					X*	
H	0.657	0.720	0.646						X*

Table 24. QCIF: Coding Only Category

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6	G7	G8
PSNR	0.708	0.691	0.635						X	X*	X
A	0.877	0.470	0.439	X*							
B	0.854	0.513	0.474		X	X*	X				
C	0.838	0.534	0.520			X	X*	X			
D	0.881	0.464	0.426	X*							
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**
E	0.831	0.543	0.506				X	X*			
F	0.859	0.502	0.475		X*	X					
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**
G	0.733	0.664	0.618						X*	X	
H	0.696	0.702	0.637							X	X*

Table 25. QCIF: Transmission Errors Category

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6	G7
PSNR	0.693	0.675	0.652				X	X*	X	
A	0.791	0.571	0.537	X*	X					
B	0.721	0.651	0.617			X	X*	X		
C	0.772	0.595	0.539	X	X*	X				
D	0.740	0.630	0.543		X	X*	X			
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
E	0.764	0.603	0.625	X	X*	X				
F	0.791	0.572	0.602	X*	X					
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
G	0.643	0.714	0.615					X	X*	X
H	0.599	0.747	0.660						X	X*

Table 26 presents a further breakdown of the RMSE of each video quality model for clips that contain coding-only impairments. The column "All Codecs" contains the model's RMSE for all 1065 video clips that contained coding-only impairments. The columns MPEG-4, H.264, VC-1, and RV-10 each contain the model's RMSE for that codec. The column "Other" contains all other codecs as previously described. The last row of the table identifies the number of video clips that fall into each subset.

Simplified group rankings are presented via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for the set of video sequences in that column (e.g., yellow in Table 26, column "All Codecs," is identical to the models identified with an "X" in Table 24, column G1). FR model cells highlighted in blue identify models that are statististically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statististically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 27 presents a further breakdown of the RMSE of each video quality model for clips that contain coding plus transmission errors. The last column of Table 27 is empty because there are too few samples to obtain a reliable estimate of RMSE (there are only 16 "Other" clips).

RMSE is used for these tables because it allows comparisons both down columns and across rows. Correlation is sensitive to the amount of variance in the set of clips associated with each column, thus numbers from different columns would not be directly comparable. RMSE allows all values in Table 26 and Table 27 to be compared.

Table 26. QCIF: Model RMSE by Codec for the Coding Only Category

FR Models	**All Codecs**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
PSNR	0.691	0.752	0.676	0.593	0.612	0.772
A	0.470	0.506	0.455	0.374	0.485	0.516
B	0.513	0.535	0.498	0.585	0.456	0.493
C	0.534	0.555	0.510	0.473	0.480	0.710
D	0.464	0.444	0.442	0.554	0.556	0.416
RR Models	**All Codecs**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
E	0.543	0.573	0.533	0.551	0.523	0.526
F	0.502	0.513	0.512	0.533	0.451	0.476
NR Models	**All Codecs**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
G	0.664	0.544	0.764	0.535	0.683	0.815
H	0.702	0.725	0.694	0.547	0.651	0.919
# of Clips	**1065**	**360**	**387**	**117**	**113**	**88**

Table 27. QCIF: Model RMSE by Codec for the Transmission Errors Category

FR Models	All Errors	MPEG-4	H.264	VC-1	RV-10	Other
PSNR	0.675	0.743	0.632	0.549	0.808	—
A	0.571	0.586	0.489	0.471	0.961	—
B	0.651	0.523	0.547	0.641	1.288	—
C	0.595	0.531	0.532	0.492	0.993	—
D	0.630	0.516	0.560	0.619	1.195	—
RR Models	**All Errors**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
E	0.603	0.558	0.559	0.620	0.847	—
F	0.572	0.532	0.495	0.601	0.803	—
NR Models	**All Errors**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
G	0.714	0.532	0.825	0.782	0.913	—
H	0.747	0.697	0.791	0.772	0.820	—
# of Clips	**751**	**280**	**199**	**192**	**64**	**16**

Table 28 identifies whether or not a model's RMSE performance is the same, better, or worse for the Transmission Errors category versus the Coding Only category. These numbers are calculated by performing an F-test on the values in Table 27 with respect to the corresponding values in Table 26. The null hypothesis would be that the Coding Only impairments and the Transmission Errors impairments are drawn from the same population. Thus, in this table "better" means that the model performed better on transmission error impairments than on coding only impairments; and "worse" means that the model performed worse on transmission error impairments than on coding only impairments. The justification for this significance test is that the response scale is the same for each pair of tests, as are the labs from which samples are drawn, the software/hardware used for HRC creation, and the source scenes used.

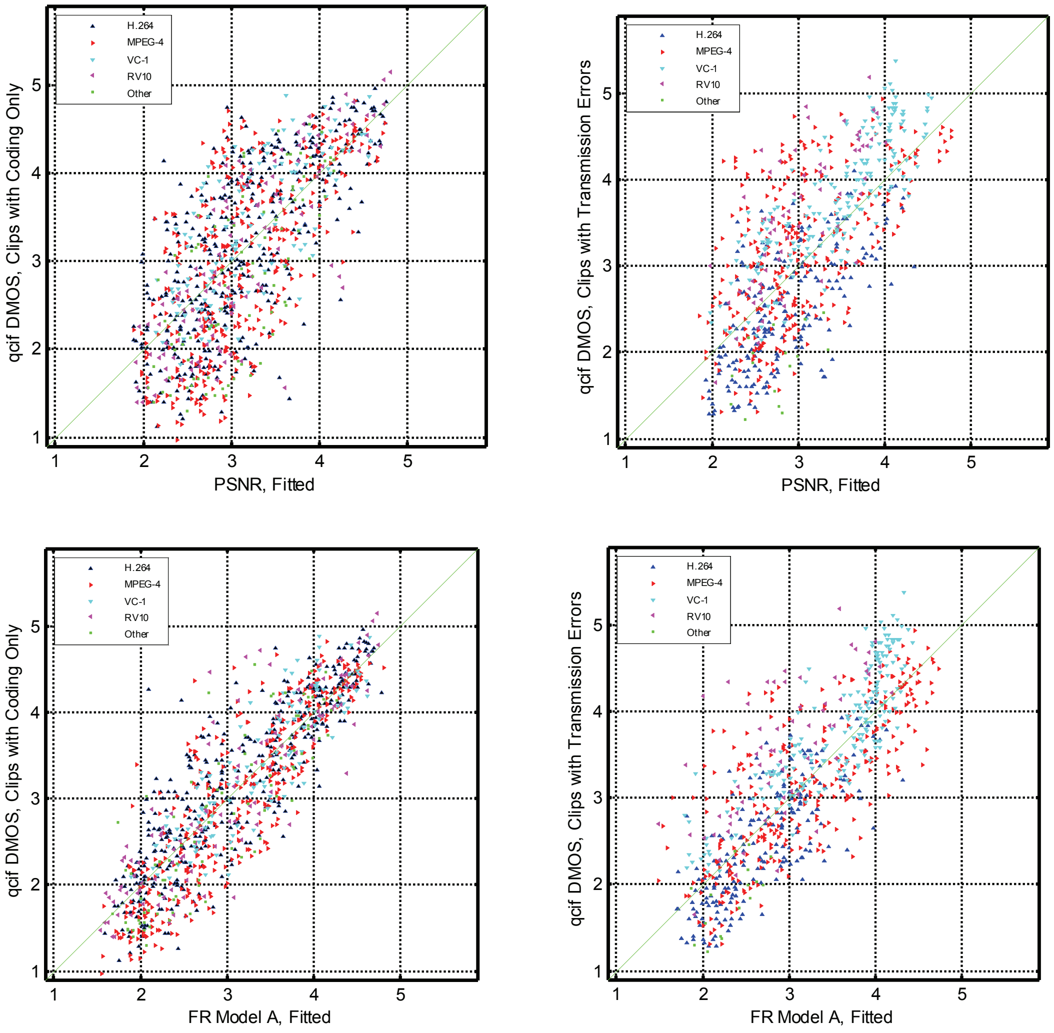
Nonetheless, caution should be used when interpreting these values. The experiments were not designed to address this question, and imbalances may exist which reduce the reliability of this assessment.

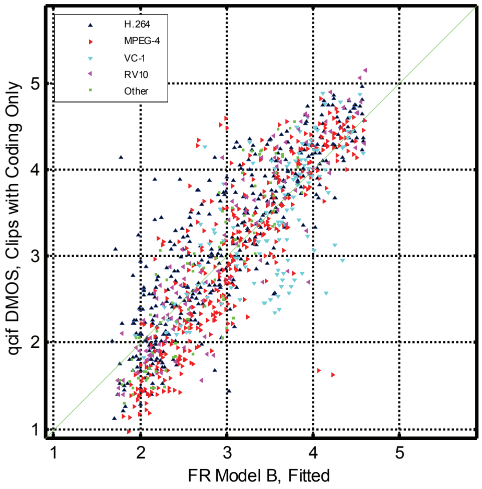
Table 28. QCIF RMSE: Transmission Errors vs. Coding Only — Same, Better, or Worse?

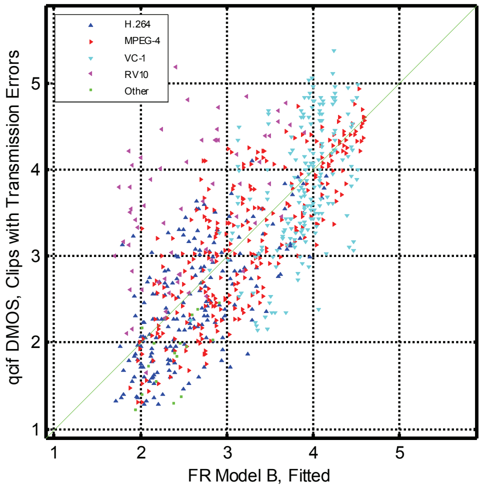
FR Models	All Errors	MPEG-4	H.264	VC-1	RV-10	Other
PSNR	Same	Same	Same	Same	Worse	—
A	Worse	Worse	Same	Worse	Worse	—
B	Worse	Same	Same	Same	Worse	—
C	Worse	Same	Same	Same	Worse	—
D	Worse	Worse	Worse	Same	Worse	—
RR Models	**All Errors**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
E	Worse	Same	Same	Same	Worse	—
F	Worse	Same	Same	Same	Worse	—
NR Models	**All Errors**	**MPEG-4**	**H.264**	**VC-1**	**RV-10**	**Other**
G	Worse	Same	Same	Worse	Worse	—
H	Worse	Same	Worse	Worse	Worse	—

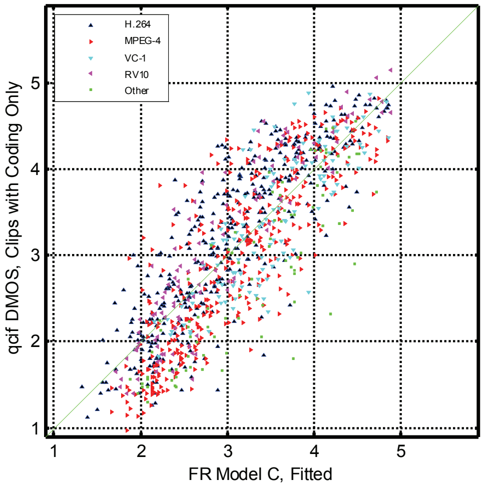
Figure 4 shows scatter plots of each model, with the QCIF Superset DMOS on the y-axis, and the fitted model score on the x-axis. There are two adjacent plots for each model, where the left hand plot contains clips with coding only artifacts, and the right hand plot contains clips with coding plus transmission errors.

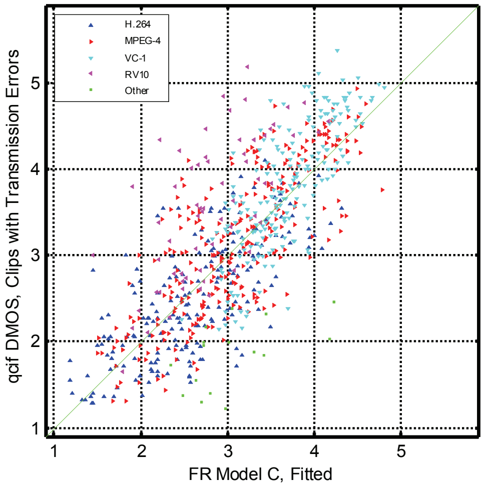
Figure 4. QCIF: coding only – left, transmission errors – right.

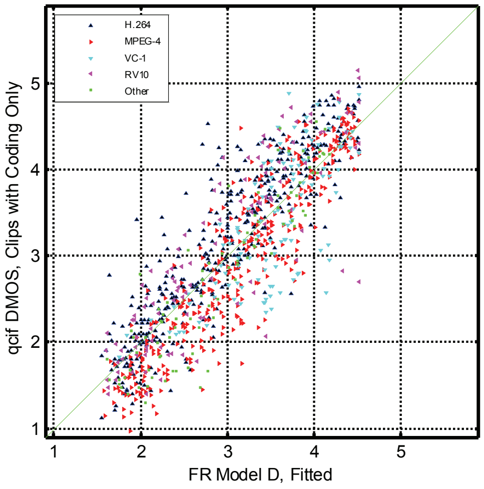


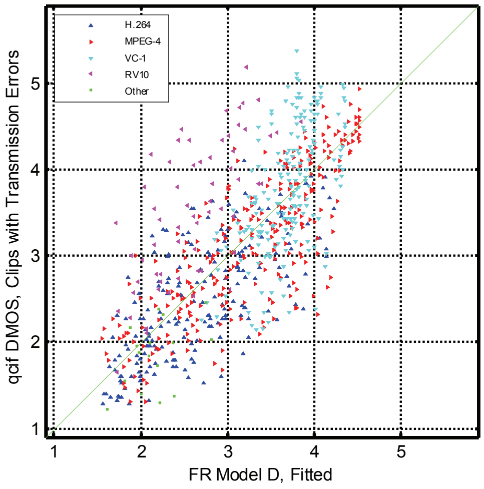

H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Coding Only
FR Model B, Fitted

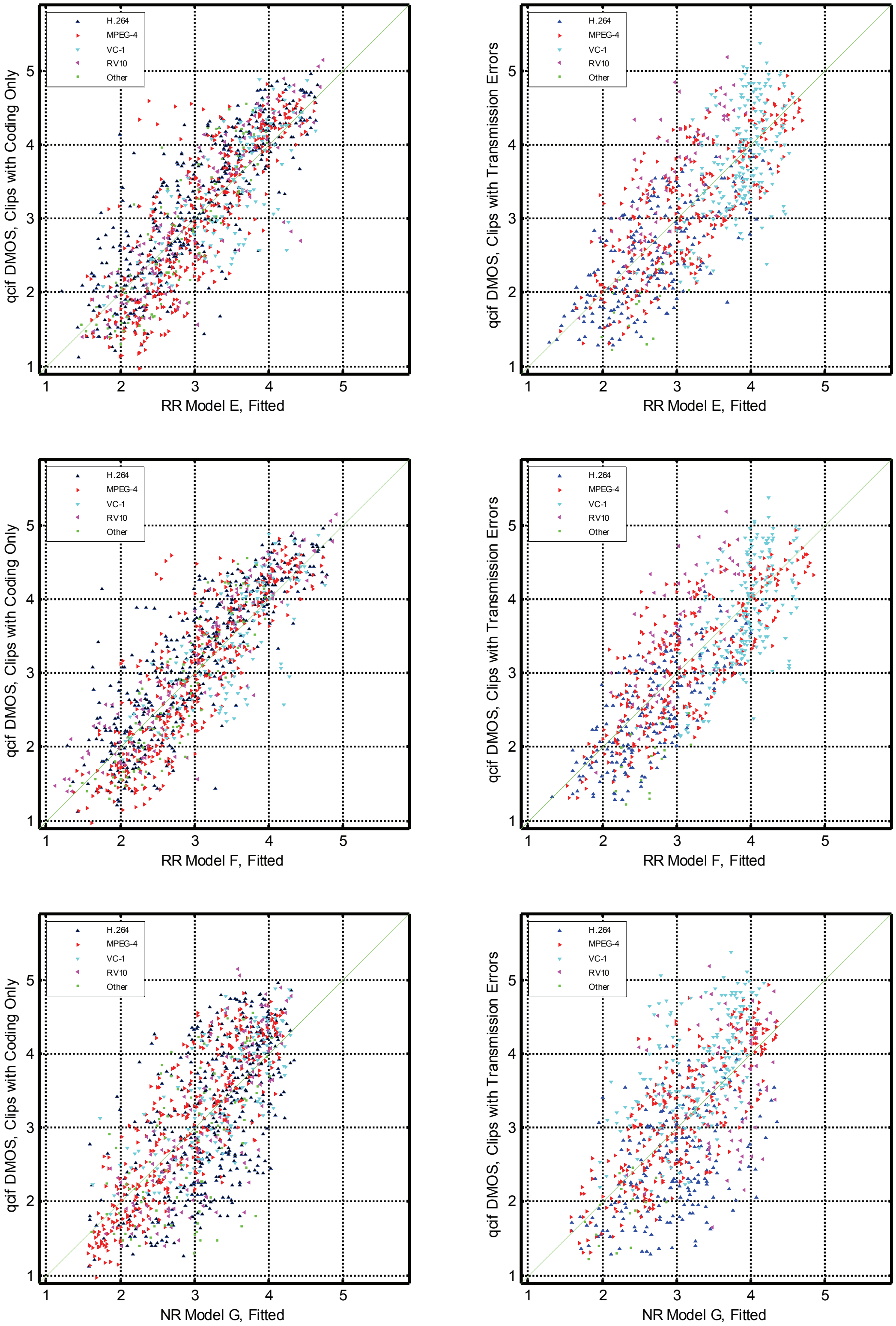

H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Transmission Errors
FR Model B, Fitted

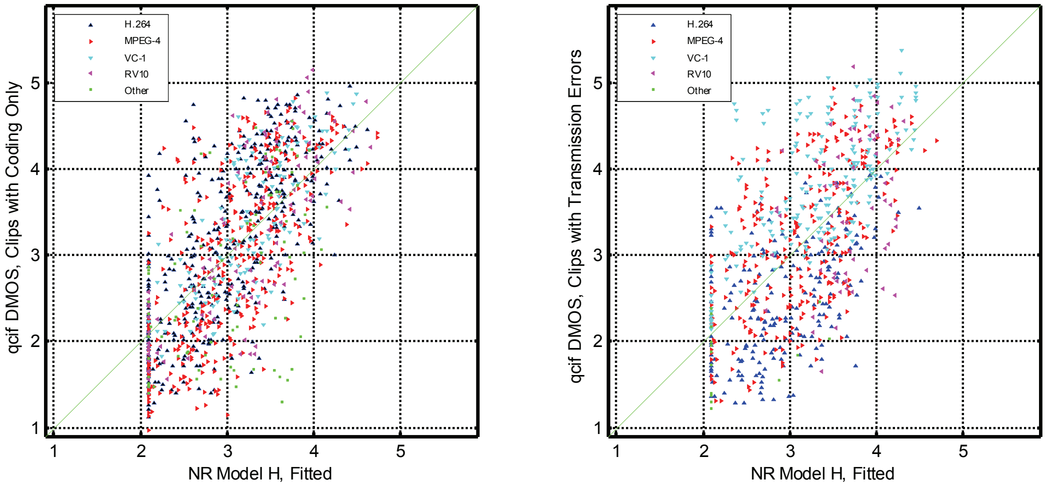

H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Coding Only
FR Model C, Fitted


H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Transmission Errors
FR Model C, Fitted


H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Coding Only
FR Model D, Fitted


H.264
MPEG-4
VC-1
RV10
Other
1
2
3
4
5
qcif DMOS, Clips with Transmission Errors
FR Model D, Fitted


H.264
MPEG-4
VC-1
RV10
Other
qcif DMOS, Clips with Coding Only
qcif DMOS, Clips with Transmission Errors
RR Model E, Fitted
RR Model F, Fitted
NR Model G, Fitted
1
2
3
4
5



Our interpretation of the QCIF results is as follows:

- The use of the models for measuring transmission error performance should be limited to MPEG-4, H.264, and VC-1 video systems. The RMSE results for RV-10 with transmission errors seem to be much worse (see Table 27). This demonstrates the danger of applying the models to other untested codecs with transmission errors. Too few other codecs (see 2nd paragraph of Section 6.1 and Table 22) were tested with transmission errors for any conclusions to be reached.

- FR model A appears to have the best overall performance (i.e., “All Codecs” column in Table 26 and “All Errors” column in Table 27). FR model A is in the group of top performing models for H.264, VC-1, and RV-10 both with and without transmission errors, and performs better than PSNR for MPEG-4 and other codecs.

- FR model C appears to be appropriate for use in analyzing video clips with transmission errors. FR model C is in the top performing group of models for transmission errors in MPEG-4, H.264, VC-1, and RV-10. This model’s performance is statistically equivalent for transmission errors versus coding only for MPEG-4, H.264, and VC-1 (see Table 27 and Table 28).

- FR model D appears to have the best performance for analyzing MPEG-4 with coding only and transmission errors (see Table 26 and Table 27).

- RR model F is at least as accurate as PSNR in all coding only and transmission error categories (see Table 26 and Table 27).

- NR model H is at least as accurate as PSNR for all coding only categories (see Table 26).

- FR models A, B, & D, and RR models E & F might be extensible for use with coding only impairments other than MPEG-4, H.264, VC-1, and RV-10 (see the “Other” column in Table 26). For this case, the other models have much worse RMSEs.

- The RMSE of NR model G is lower for MPEG-4 (both coding only and transmission errors) and VC-1 (coding only) than for the other conditions (look across the row for model G in Table 26 and Table 27).

- The RMSE of NR model H is lower for VC-1 (coding only) than the other conditions (look across the row for model H in Table 26).

- The analysis is poorly balanced with respect to codec type. Had this type of analysis been a primary goal of the experiments, then the experiments would have been designed to have an approximately equal number of clips associated with each codec (see Table 22).

6.2 CIF Results

This section examines CIF model performance for two major subdivisions of the video clips: video clips with only coding artifacts, and video clips with coding artifacts plus transmission errors. For each major category, a further subdivision is examined to determine how the models perform with respect to codec type. Codec types have been divided into the following five sub-categories: H.264, MPEG-4 (excluding H.264), Video Codec 1 (VC-1, also known as Windows Media 9), Real Video 10 (RV-10), and Other.

The “Other” sub-category includes a variety of codecs each used for one to four HRCs: JPEG-2000 (4 HRCs), H.261 (1 HRC), MPEG-1 (2 HRCs & 2 common clips), H.263 (3 HRCs), DivX (2 HRCs & 2 common clips), Sorenson (1 HRC), and Cinepak (1 HRC). Some of these “Other” codecs were created using proprietary codecs, and some utilized non-standard variations of the mentioned codecs.

Table 29 shows the number of CIF video clips associated with each major category. Note that codecs are not evenly balanced (i.e., different number of clips) with respect to the “Coding Only” and “Transmission Errors” categories. Thus, when multiple codecs are combined, this will skew results more heavily toward those systems that have more clips. These imbalances are also present in the global analysis presented in Section 5.2.

Table 29. CIF: Number of Clips with Each Type of Impairment

CODEC	CODING ONLY	TRANSMISSION ERRORS
H.264	465	278
MPEG-4	312	240
VC-1	108	0
RV-10	160	64
Other	149	0
TOTAL	1234	582

Table 30 lists the following statistics, computed using all the video sequences in the CIF superset: Pearson correlation, RMSE, outlier ratio, and the ranking groups computed using RMSE. The information in Table 30 is identical to that presented in Table 13, but is reproduced here for easy comparisons. Table 31 repeats this analysis, but uses only those video sequences that contain coding artifacts. Table 32 uses those video sequences that contain transmission errors.

Table 30. CIF: All Video Sequences

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.642	0.735	0.671				X*	
I	0.794	0.582	0.539		X*			
J	0.759	0.624	0.567			X*		
K	0.847	0.509	0.507	X*				
L	0.795	0.582	0.550		X*			
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.773	0.607	0.569			X*		
N	0.773	0.607	0.566			X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.478	0.841	0.698					X*
P	0.516	0.821	0.688					X*

Table 31. CIF: Coding Only Category

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6
PSNR	0.646	0.768	0.690					X*	
I	0.843	0.545	0.497		X*				
J	0.771	0.643	0.579				X*		
K	0.873	0.495	0.502	X*					
L	0.826	0.572	0.553			X*			
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
M	0.811	0.591	0.546			X*			
N	0.810	0.593	0.545			X*			
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
O	0.527	0.864	0.707						X*
P	0.586	0.825	0.691						X*

Table 32. CIF: Transmission Errors Category

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6
PSNR	0.617	0.660	0.631				X	X*	
I	0.637	0.656	0.629				X	X*	
J	0.719	0.585	0.540		X*	X			
K	0.784	0.539	0.517	X*					
L	0.719	0.603	0.541		X	X*	X		
RR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
M	0.653	0.641	0.617			X	X*	X	
N	0.656	0.639	0.610			X	X*	X	
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
O	0.417	0.794	0.679						X*
P	0.365	0.814	0.680						X*

Table 33 presents a further breakdown of the RMSE of each video quality model for clips that contain coding only impairments. The column "All Codecs" contains the model's RMSE for all 1234 video clips that contained coding only impairments. The columns MPEG-4, H.264, VC-1, and RV-10 each contain the model's RMSE for that codec. The column "Other" contains all other codecs as previously described. The last row of the table identifies the number of video clips that fall into each subset.

Simplified group rankings are presented via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for the set of video sequences in that column (e.g., yellow in Table 33, column "All Codecs," is identical to the models identified with an "X" in Table 31, column G1). FR model cells highlighted in blue identify models that are statististically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statistically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 34 presents a further breakdown of the RMSE of each video quality model for clips that contain coding plus transmission errors. The last two columns of Table 34 are empty because there are no samples.

RMSE is used for these tables because it allows comparisons both down columns and across rows. Correlation is sensitive to the amount of variance in the set of clips associated with each column, thus numbers from different columns would not be directly comparable. RMSE allows all values in Table 33 and Table 34 to be compared.

Table 33. CIF: Model RMSE by Codec for the Coding Only Category

FR Models	**All Codecs**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
PSNR	0.768	0.829	0.780	0.685	0.776	0.624
I	0.545	0.570	0.553	0.491	0.599	0.493
J	0.643	0.644	0.611	0.557	0.859	0.597
K	0.495	0.485	0.454	0.545	0.608	0.481
L	0.572	0.554	0.582	0.670	0.556	0.560
RR Models	**All Codecs**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
M	0.591	0.634	0.617	0.529	0.595	0.480
N	0.593	0.635	0.620	0.531	0.595	0.481
NR Models	**All Codecs**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
O	0.864	0.893	0.753	0.986	0.914	0.943
P	0.825	0.881	0.722	0.880	0.862	0.839
# of Clips	**1234**	**465**	**312**	**160**	**149**	**108**

Table 34. CIF: Model RMSE by Codec for the Transmission Errors Category

FR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
PSNR	0.660	0.686	0.665	0.555	—	—
I	0.656	0.640	0.630	0.848	—	—
J	0.585	0.647	0.514	0.585	—	—
K	0.539	0.580	0.495	0.545	—	—
L	0.603	0.650	0.521	0.712	—	—
RR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
M	0.641	0.664	0.649	0.539	—	—
N	0.639	0.660	0.650	0.532	—	—
NR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
O	0.794	0.778	0.807	0.864	—	—
P	0.814	0.843	0.766	0.911	—	—
# of Clips	**582**	**278**	**240**	**64**	**0**	**0**

Table 35 identifies whether or not a model's RMSE performance is the same, better, or worse for the Transmission Errors category versus the Coding Only category. These numbers are calculated by performing an F-test on the values in Table 34 with respect to the corresponding values in Table 33. The null hypothesis would be that the Coding Only impairments and the Transmission Errors impairments are drawn from the same population. Thus, "better" means that the model performed better on transmission error impairments than on coding only impairments; and "worse" means that the model performed worse on transmission error impairments than on coding only impairments.The justification for this significance test is that the response scale is the same for each pair of tests, as are the labs from which samples are drawn, the software/hardware used for HRC creation, and the source scenes used.

Nonetheless, caution should be used when interpreting these values. The experiments were not designed to address this question, and imbalances may exist which reduce the reliability of this assessment.

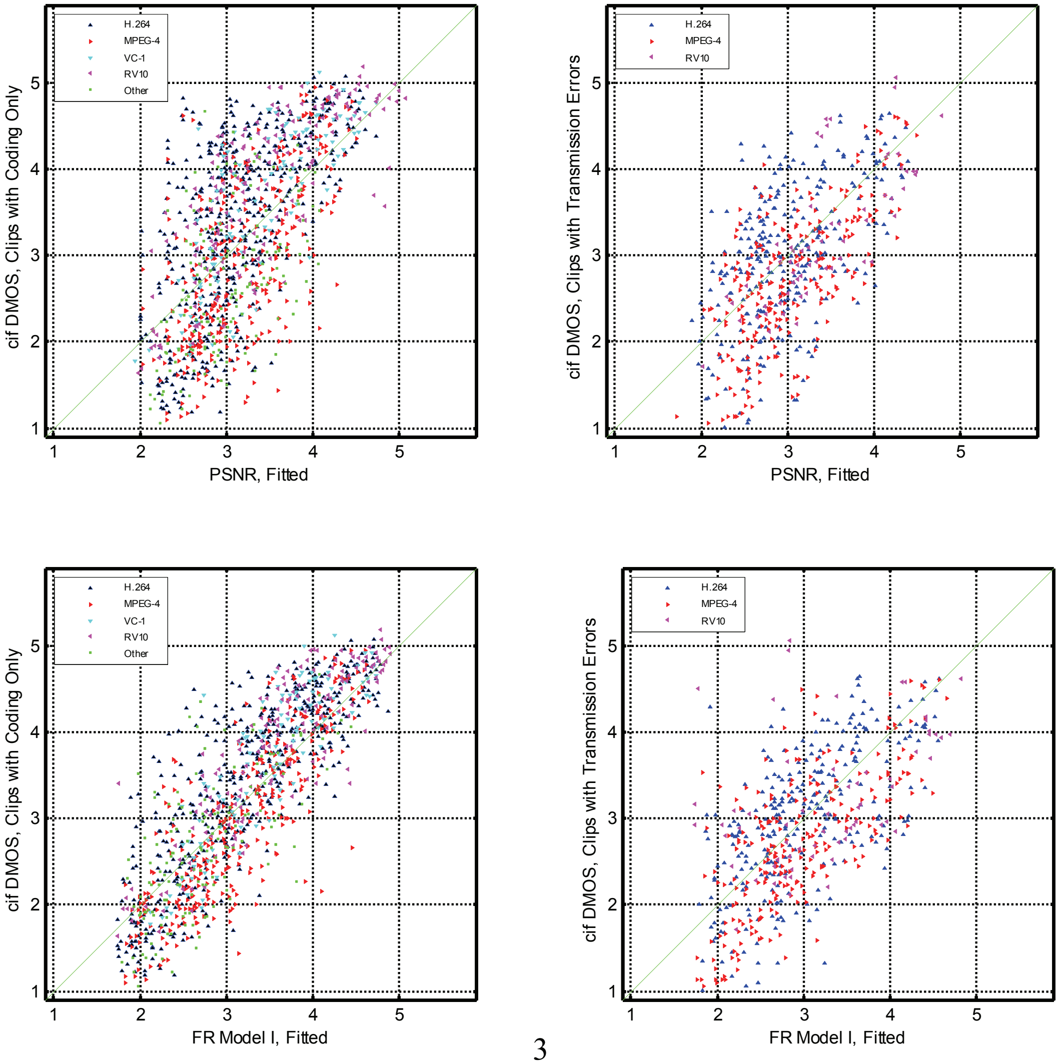
Table 35. CIF RMSE: Transmission Errors vs. Coding Only — Same, Better, or Worse?

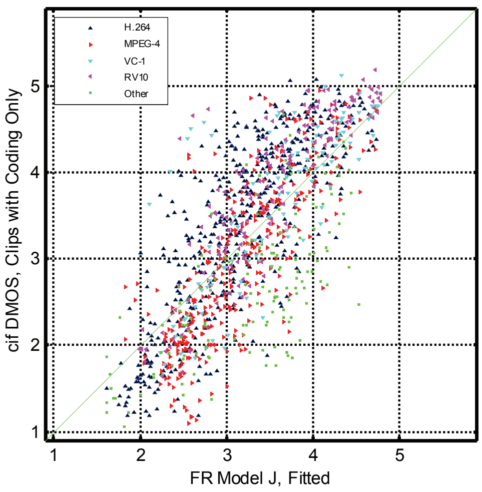
FR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
PSNR	Better	Better	Better	Better	—	—
I	Worse	Worse	Worse	Worse	—	—
J	Better	Same	Better	Same	—	—
K	Worse	Worse	Same	Same	—	—
L	Same	Worse	Better	Same	—	—
RR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
M	Worse	Same	Same	Same	—	—
N	Worse	Same	Same	Same	—	—
NR Models	**All Errors**	**H.264**	**MPEG-4**	**RV-10**	**Other**	**VC-1**
O	Better	Better	Same	Same	—	—
P	Same	Same	Same	Same	—	—

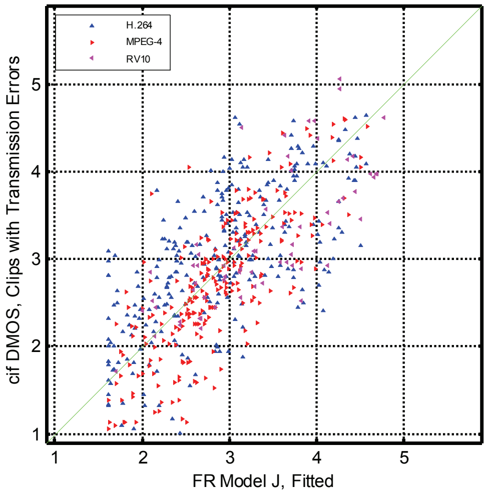
Figure 5 shows scatter plots of each model, with the CIF Superset DMOS on the y-axis, and the fitted model score on the x-axis. There are two adjacent plots for each model, where the left hand plot contains clips with coding only artifacts, and the right hand plot contains clips with coding plus transmission errors.

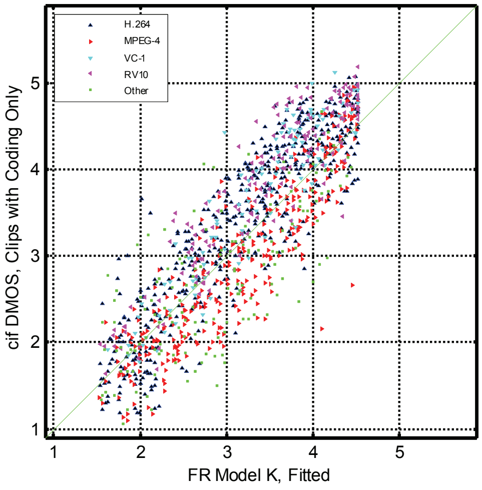
These scatter plots show that transmission errors span a more restricted range of quality when compared to coding only. To see this, compare the vertical distribution of the right-hand plots and left-hand plots. Pearson correlation is sensitive to the spread of data. This is why the drop in correlation often appears more severe than the drop in RMSE when one compares Transmission Errors with Coding Only. Note that RMSE is not sensitive to the range of quality spanned.

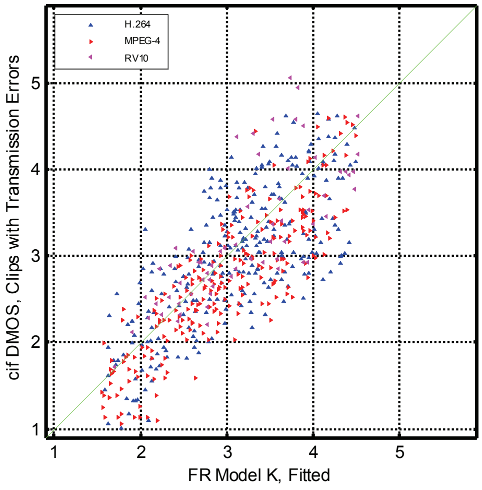
Figure 5. CIF: coding only – left, transmission errors – right.

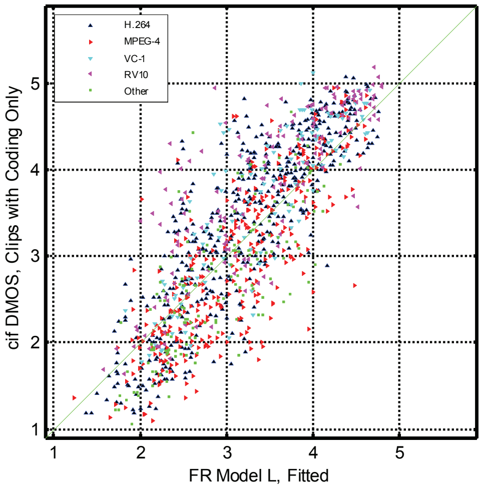


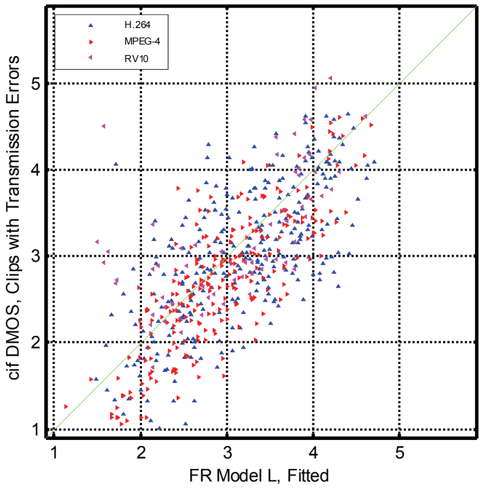

H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
FR Model J, Fitted

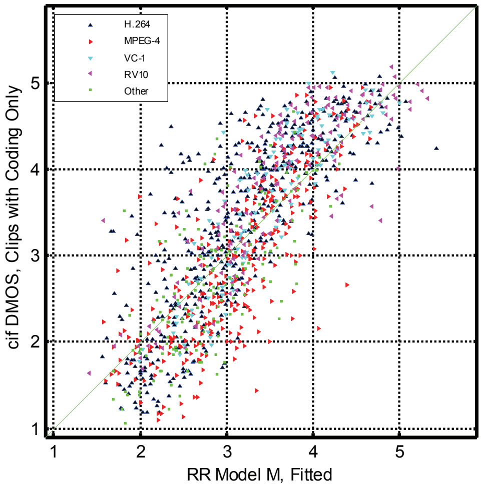

H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
FR Model J, Fitted

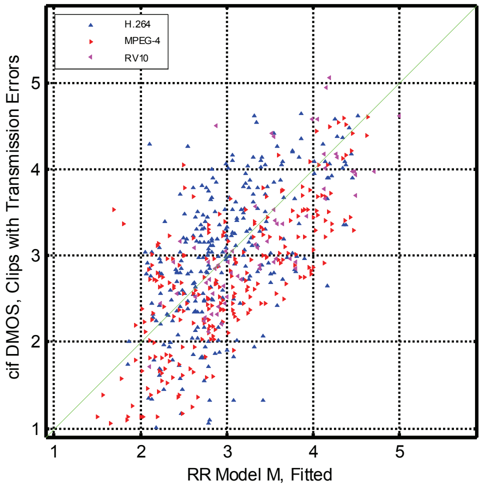

H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
FR Model K, Fitted

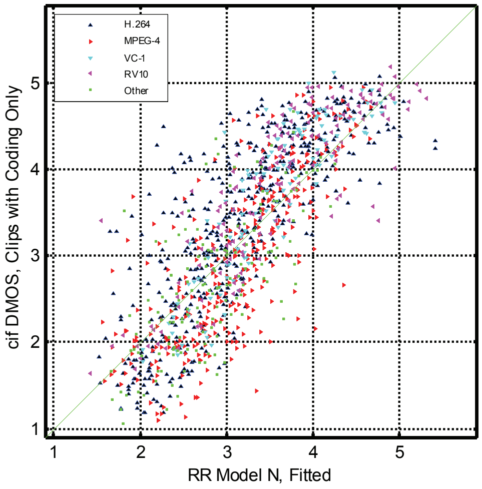

H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
FR Model K, Fitted

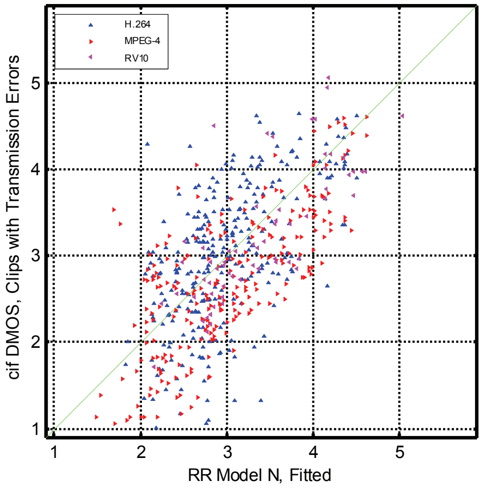

H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
FR Model L, Fitted

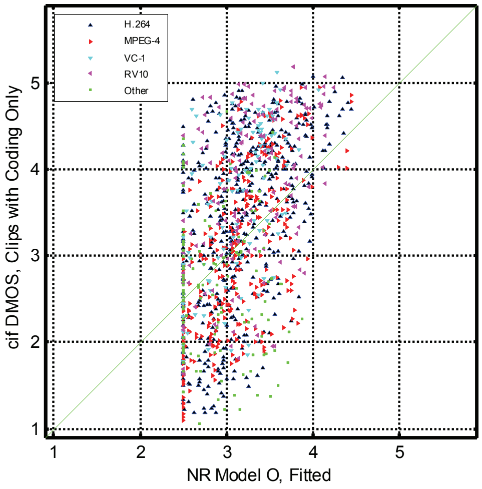

H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
FR Model L, Fitted

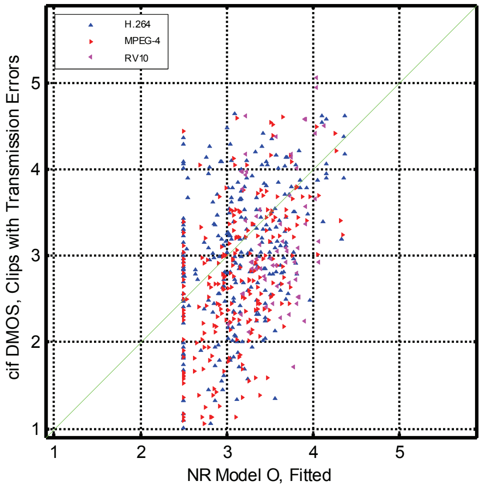

H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
RR Model M, Fitted
1
2
3
4
5

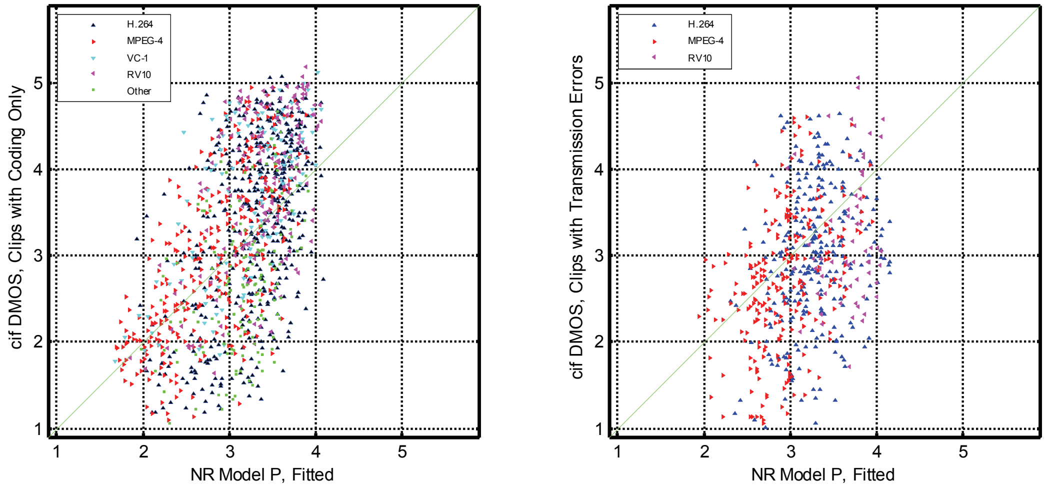

H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
RR Model M, Fitted
1
2
3
4
5


H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
RR Model N, Fitted
1
2
3
4
5


H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
RR Model N, Fitted
1
2
3
4
5


H.264
MPEG-4
VC-1
RV10
Other
cif DMOS, Clips with Coding Only
1
2
3
4
5
NR Model O, Fitted
1
2
3
4
5


H.264
MPEG-4
RV10
cif DMOS, Clips with Transmission Errors
1
2
3
4
5
NR Model O, Fitted
1
2
3
4
5



Our interpretation of the CIF results is as follows:

- The use of the models for measuring transmission error performance should be limited to MPEG-4, H.264, and RV-10 video systems. VC-1 (Windows Media 9) was never tested with transmission errors, nor were any of the "Other" codecs (see 2nd paragraph of Section 6.2 and Table 29).

- FR model K demonstrates the best overall performance (i.e., model K is in the group of top performing models for all columns in Table 33 and Table 34) both with and without transmission errors.

- RR models M and N are at least as accurate as PSNR in all coding only and transmission error categories (see Table 33 and Table 34).

- FR model I has consistently better performance for coding only impairments than for coding plus transmission errors (see Table 35, and compare Table 33 with Table 34).

- PSNR has better performance when analyzing transmission errors than when analyzing coding only impairments (compare RMSE values in Table 33 and Table 34).

- FR models I, K, & L, and RR models M & N might be extensible for use with coding only impairments other than MPEG-4, H.264, VC-1, and RV-10 (see the "Other" column in Table 33). For this case, the other models have much worse RMSEs.

- The analysis is poorly balanced with respect to codec type. Had this type of analysis been a primary goal of the experiments, then the experiments would have been designed to have an approximately equal number of clips associated with each codec (see Table 29).

6.3 VGA Results

This section examines VGA model performance for two major subdivisions of the video clips: video clips with only coding artifacts, and video clips with coding artifacts plus transmission errors. For each major category, a further subdivision is examined to determine how the models perform with respect to codec type. Codec types have been divided into the following five sub-categories: H.264, MPEG-4 (excluding H.264), Video Codec 1 (VC-1, also known as Windows Media 9), Real Video 10 (RV-10), and Other.

The "Other" category includes a variety of codecs each used for one to six HRCs: JPEG-2000 (2 HRCs), H.261 (3 HRCs), MPEG-2 (6 HRCs & 1 common clip), H.263 (2 HRCs + 2 common clips), SVC (5 HRCs), DivX (2 HRCs), Sorenson (1 HRC + 2 common clips), Cinepak (1 HRC), and Theora (1 HRC). Some of these "Other" codecs were created using proprietary codecs, and some utilized non-standard variations of the mentioned codecs.

Table 36 shows the number of video clips associated with each major category. Note that codecs are not evenly balanced (i.e., different number of clips) with respect to the "Coding Only" and "Transmission Errors" categories. Thus, when multiple codecs are combined, this will skew results more heavily toward those systems that have more clips. These imbalances are also present in the global analysis presented in Section 5.3.

Table 36. VGA: Number of Video Clips in Each Category

CODEC	CODING ONLY	TRANSMISSION ERRORS
H.264	678	108
MPEG-4	235	154
VC-1	93	0
RV-10	136	64
Other	171	25
TOTAL	1313	351

Table 37 lists the following statistics, computed using all the video sequences in the VGA superset: Pearson correlation, RMSE, outlier ratio, and the ranking groups computed using RMSE. The information in Table 37 is identical to that presented in Table 18, but is reproduced here for easy comparisons. Table 38 repeats this analysis, but uses only those video sequences that contain coding artifacts. Table 39 uses those video sequences that contain transmission errors.

Table 37. VGA: All Video Sequences

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.727	0.701	0.638				X*	
Q	0.818	0.586	0.556	X*	X			
R	0.741	0.686	0.624				X*	
S	0.803	0.608	0.558	X	X*	X		
T	0.797	0.617	0.564		X	X*		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
U	0.799	0.613	0.575		X	X*		
V	0.800	0.613	0.579		X	X*		
W	0.800	0.612	0.578		X	X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
X	0.408	0.932	0.751					X*
Y	0.429	0.922	0.722					X*

Table 38. VGA: Coding Only Category

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
PSNR	0.743	0.692	0.636					X*	
Q	0.831	0.576	0.560	X	X*	X			
R	0.759	0.673	0.623					X*	
S	0.848	0.552	0.534	X*	X				
T	0.833	0.572	0.545	X*	X				
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
U	0.812	0.604	0.564			X	X*		
V	0.812	0.603	0.569			X	X*		
W	0.813	0.602	0.567		X	X*	X		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
X	0.406	0.948	0.762						X*
Y	0.473	0.914	0.727						X*

Table 39. VGA: Transmission Errors Category

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
PSNR	0.604	0.738	0.644		X*		
Q	0.742	0.629	0.544	X*			
R	0.611	0.737	0.630		X*		
S	0.567	0.787	0.650		X*		
T	0.591	0.763	0.638		X*		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
U	0.722	0.652	0.618	X*			
V	0.721	0.652	0.618	X*			
W	0.720	0.653	0.618	X*			
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
X	0.492	0.874	0.709			X*	
Y	0.236	0.957	0.707				X*

Table 40 presents a further breakdown of the RMSE of each video quality model for clips that contain coding only impairments. The column “All Codecs” contains the model’s RMSE for all 1313 video clips that contained coding only impairments. The columns MPEG-4, H.264, VC-1, and RV-10 each contain the model’s RMSE for that codec. The column “Other” contains all other codecs as previously described. The last row of the table identifies the number of video clips that fall into each subset.

Simplified group rankings are presented via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for the set of video sequences in that column (e.g., yellow in Table 40, column “All Codecs,” is identical to the models identified with an “X” in Table 38, column G1). FR model cells highlighted in blue identify models that are statististically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statististically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 41 presents a further breakdown of the RMSE of each video quality model for clips that contain coding plus transmission errors. Two columns of Table 41 are empty because there are too few samples to obtain a reliable estimate of RMSE (there are only 25 “Other” clips and no VC-1 clips).

RMSE is used for these tables because it allows comparisons both down columns and across rows. Correlation is sensitive to the amount of variance in the set of clips associated with each column, thus numbers from different columns would not be directly comparable. RMSE allows all values in Table 40 and Table 41 to be compared.

Table 40. VGA: Model RMSE by Codec for the Coding Only Category

FR Model	All Codecs	H.264	MPEG-4	Other	RV-10	VC-1
PSNR	0.692	0.728	0.587	0.790	0.623	0.617
Q	0.576	0.645	0.438	0.534	0.539	0.511
R	0.673	0.714	0.528	0.835	0.605	0.473
S	0.552	0.564	0.515	0.642	0.544	0.410
T	0.572	0.588	0.455	0.679	0.629	0.458
RR Model	**All Codecs**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
U	0.604	0.646	0.487	0.714	0.553	0.417
V	0.603	0.646	0.484	0.715	0.554	0.417
W	0.602	0.645	0.482	0.713	0.552	0.416
NR Model	**All Codecs**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
X	0.948	0.988	0.853	0.974	0.939	0.927
Y	0.914	0.880	0.980	1.014	0.942	0.819
# of Clips	**1313**	**678**	**235**	**171**	**136**	**93**

Table 41. VGA: Model RMSE by Codec for the Transmission Errors Category

FR Models	All Errors	H.264	MPEG-4	Other	RV-10	VC-1
PSNR	0.738	0.702	0.771	—	0.702	—
Q	0.629	0.661	0.558	—	0.762	—
R	0.737	0.834	0.688	—	0.614	—
S	0.787	0.732	0.859	—	0.790	—
T	0.763	0.645	0.759	—	0.998	—
RR Model	**All Errors**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
U	0.652	0.644	0.626	—	0.695	—
V	0.652	0.647	0.625	—	0.696	—
W	0.653	0.645	0.622	—	0.707	—
NR Models	**All Errors**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
X	0.874	0.936	0.870	—	0.719	—
Y	0.957	1.057	0.804	—	0.847	—
# of Clips	**351**	**108**	**154**	**25**	**64**	**0**

Table 42 identifies whether or not a model's RMSE performance is the same, better, or worse for the Transmission Errors category versus the Coding Only category. These numbers are calculated by performing an F-test on the values in Table 41 with respect to the corresponding values in Table 40. The null hypothesis would be that the Coding Only impairments and the Transmission Errors impairments are drawn from the same population. Thus, "better" means that the model performed better on transmission error impairments than on coding only impairments; and "worse" means that the model performed worse on transmission error impairments than on coding only impairments.The justification for this significance test is that the response scale is the same for each pair of tests, as are the labs from which samples are drawn, the software/hardware used for HRC creation, and the source scenes used.

Nonetheless, caution should be used when interpreting these values. The experiments were not designed to address this question, and imbalances may exist which reduce the reliability of this assessment.

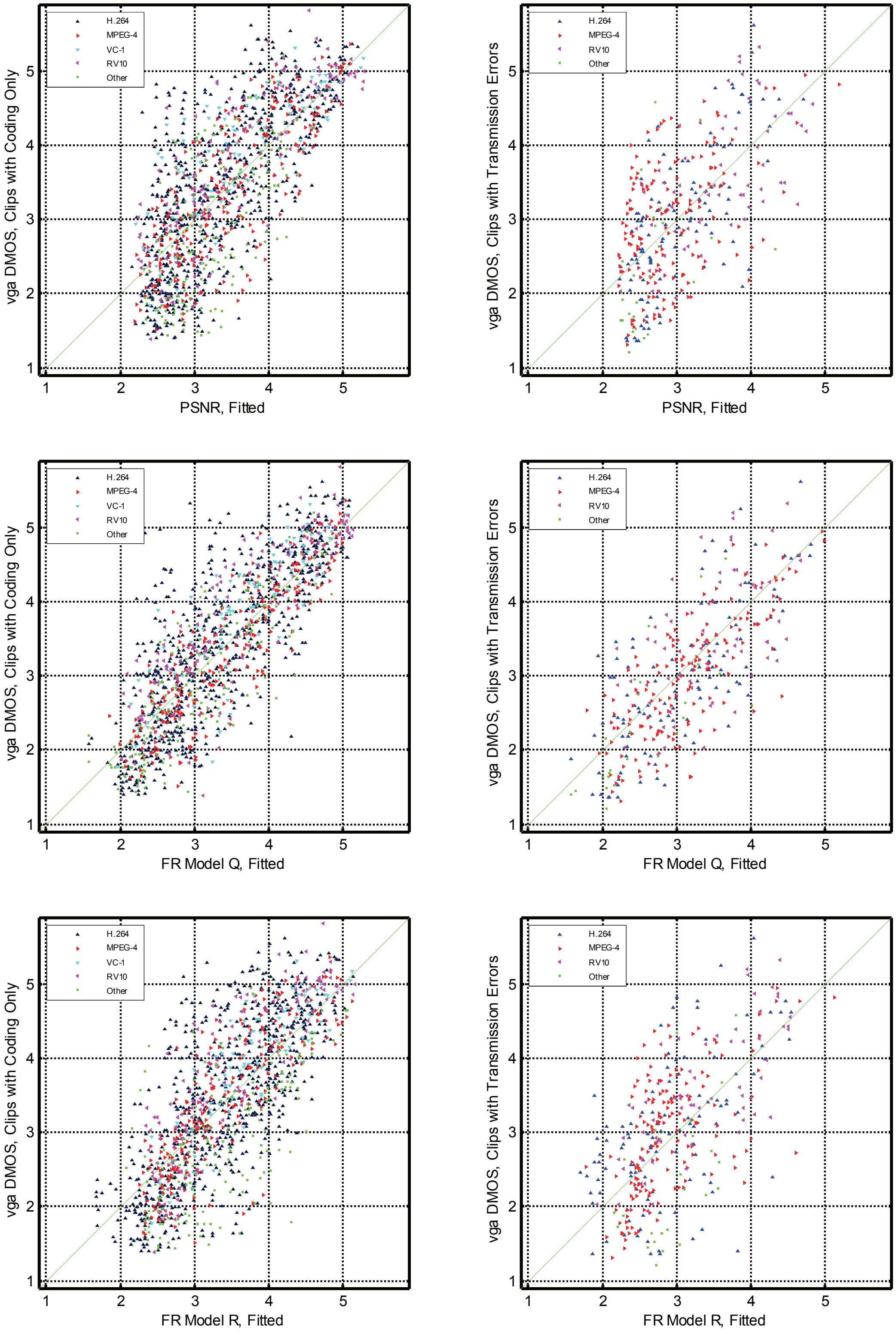
Table 42. VGA RMSE: Transmission Errors vs. Coding Only — Same, Better, or Worse?

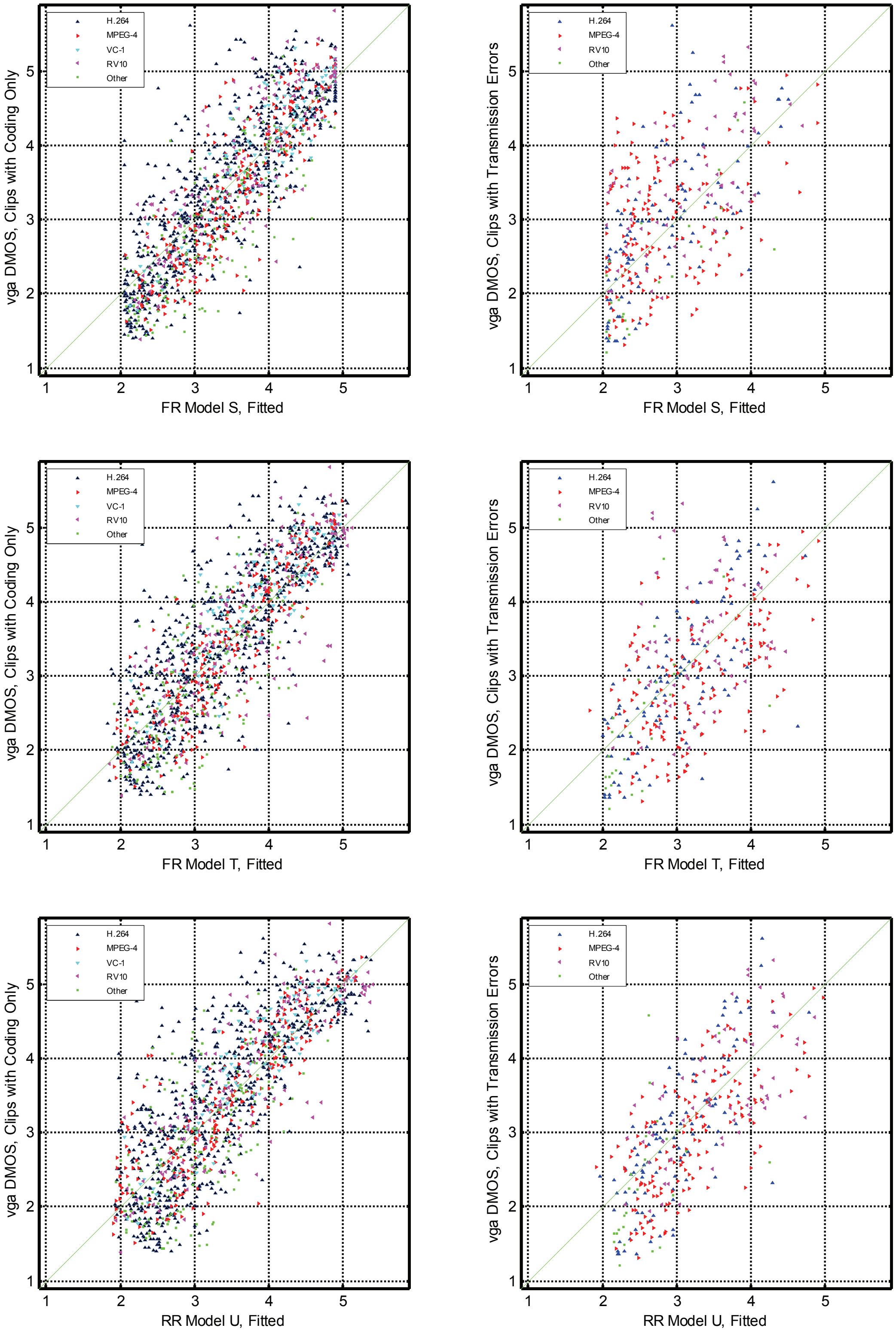
FR Models	**All Errors**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
PSNR	Same	Same	Worse	—	Same	—
Q	Worse	Same	Worse	—	Worse	—
R	Worse	Worse	Worse	—	Same	—
S	Worse	Worse	Worse	—	Worse	—
T	Worse	Same	Worse	—	Worse	—
RR Model	**All Errors**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
U	Worse	Same	Worse	—	Worse	—
V	Worse	Same	Worse	—	Worse	—
W	Worse	Same	Worse	—	Worse	—
NR Models	**All Errors**	**H.264**	**MPEG-4**	**Other**	**RV-10**	**VC-1**
X	Better	Same	Same	—	Better	—
Y	Same	Worse	Better	—	Same	—

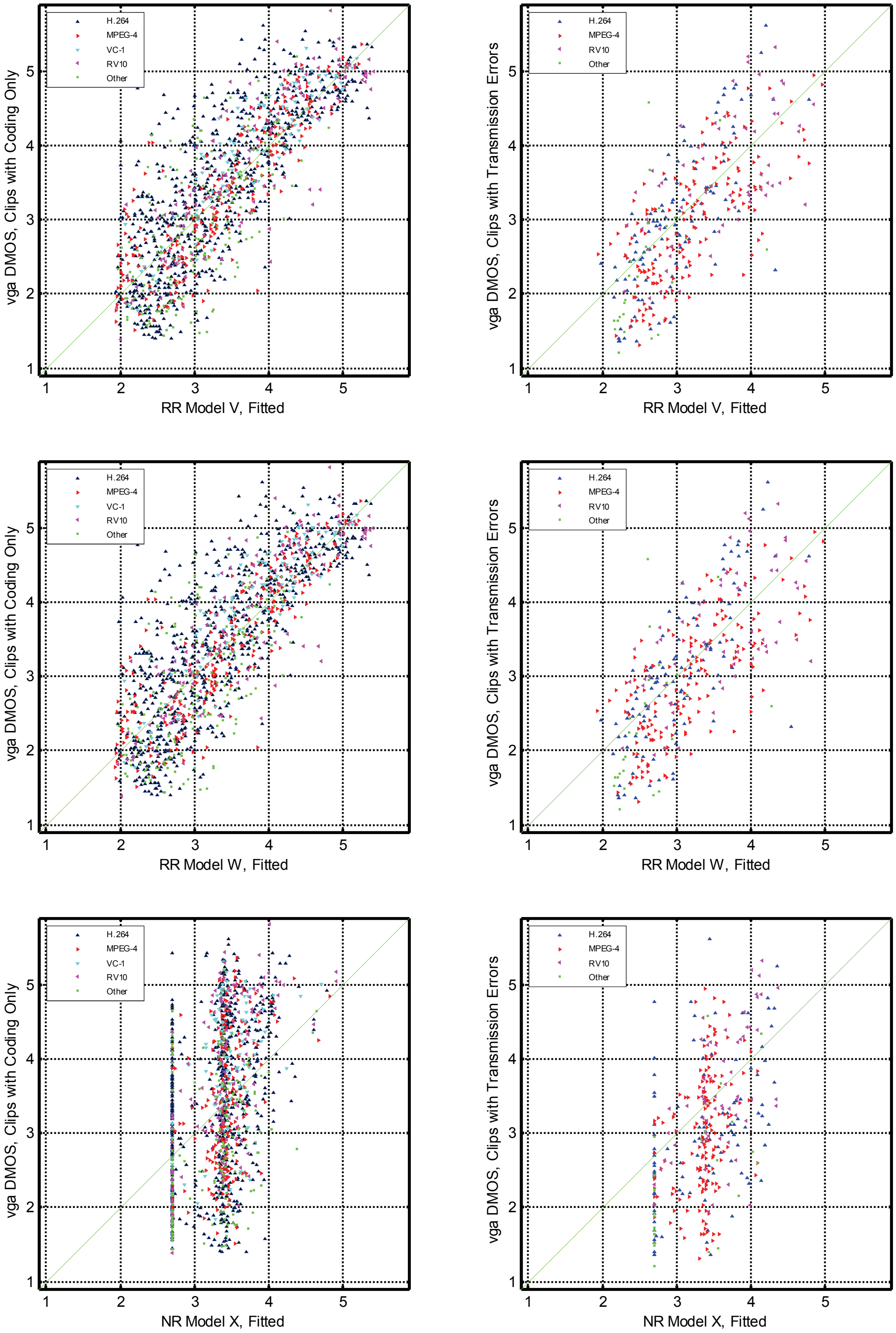
Figure 6 shows scatter plots of each model, with the VGA Superset DMOS on the y-axis, and the fitted model score on the x-axis. There are two adjacent plots for each model, where the left hand plot contains clips with coding only artifacts, and the right hand plot contains clips with coding plus transmission errors. Models are presented alphabetically from top to bottom.

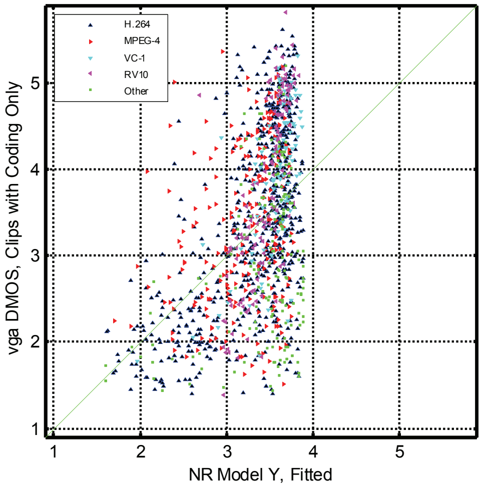
These scatter plots show that transmission errors span a more restricted range of quality when compared to coding only. To see this, compare the vertical distribution of the right-hand plots and left-hand plots. Pearson correlation is sensitive to the spread of data. This is why the drop in correlation often appears more severe than the drop in RMSE when one compares Transmission Errors with Coding Only. Note that RMSE is not sensitive to the range of quality spanned.

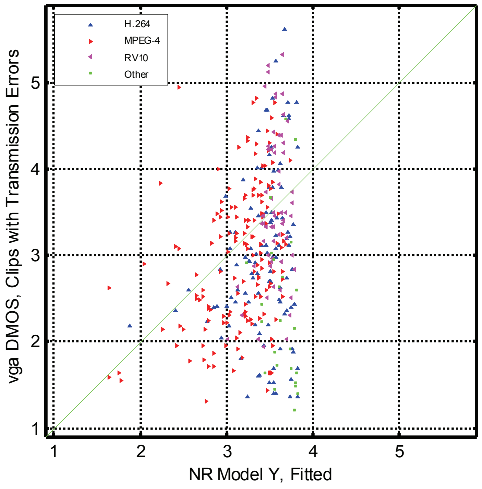
Figure 6. VGA: coding only – left, transmission errors – right.




H.264
MPEG-4
VC-1
RV10
Other
vga DMOS, Clips with Coding Only
FR Model S, Fitted
H.264
MPEG-4
RV10
Other
vga DMOS, Clips with Transmission Errors
FR Model S, Fitted
H.264
MPEG-4
VC-1
RV10
Other
vga DMOS, Clips with Coding Only
FR Model T, Fitted
H.264
MPEG-4
RV10
Other
vga DMOS, Clips with Transmission Errors
FR Model T, Fitted
H.264
MPEG-4
VC-1
RV10
Other
vga DMOS, Clips with Coding Only
RR Model U, Fitted
H.264
MPEG-4
RV10
Other
vga DMOS, Clips with Transmission Errors
RR Model U, Fitted
1
2
3
4
5


H.264
MPEG-4
VC-1
RV10
Other
vga DMOS, Clips with Coding Only
RR Model V, Fitted
vga DMOS, Clips with Transmission Errors
RR Model V, Fitted
RR Model W, Fitted
RR Model W, Fitted
NR Model X, Fitted
NR Model X, Fitted
1
2
3
4
5





Our interpretation of the VGA results is as follows:

- The use of the models for measuring transmission error performance should be limited to MPEG-4, H.264, and RV-10 video systems. VC-1 (Windows Media 9) was never tested with transmission errors, and insufficient data exists to reach any definitive conclusions about the extensibility of models to other codecs with transmission errors (see 2nd paragraph of Section 6.3 and Table 36).

- FR model Q appears to have the best overall performance (i.e., "All Codecs" column in Table 38 and "All Errors" column in Table 40). FR model Q performs better than PSNR in all categories except H.264 with transmission errors (where PSNR and model Q are both in the group of top performing models) and RV-10 with transmission errors (where PSNR performs statistically better than model Q).

- FR models S and T are the only models that are in the top performing group for both H.264 coding only and H.264 with transmission errors (see Table 40 and Table 41).

- RR models U, V, and W appear to be as appropriate for use in analyzing video clips with transmission errors as the top performing FR models. RR models U, V, and W are in the top performing group of models for all transmission error categories (i.e., columns "All Codecs," "H.264," "MPEG-4," and "RV-10" in Table 41).

- RR models U, V, and W are at least as accurate as PSNR in all coding only and transmission error categories (see Table 40 and Table 41).

- All of the FR and RR models have reduced performance for MPEG-4 (excluding part 10 aka H.264) when transmission errors are present (see Table 42).

- The analysis is poorly balanced with respect to codec type. Had this type of analysis been a primary goal of the experiments, then the experiments would have been designed to have an approximately equal number of clips associated with each codec (see Table 36).

7 ESTIMATING HRC QUALITY

This section examines how a model's accuracy changes when several scenes are passed through a single video HRC, and the scores for those scenes are averaged to produce an average quality score for the HRC. This indicates how well the objective models track the overall average HRC quality level. For this analysis, we average the results from multiple scenes to obtain an overall average quality estimate for each specific HRC. For a given HRC, all HRC settings (e.g., coder settings including any user-selectable options such as frame rate, the network properties, and the decoder settings) are fixed. Each individual MM experiment utilized eight (8) source (SRC) video sequences that were passed through 16 HRCs. When averaging over all 8 SRCs, this reduces the number of data points by a factor of 8. The common set sequences are discarded for the HRC analysis since most common set HRCs were associated with only one scene.

There have been some concerns that the results associated with the HRC analysis performed in the above fashion might be better than what an end-user would achieve in the field. The possible reason for this is that the SRCs used in the MM experiments were chosen very carefully, and balanced using a carefully chosen set of criteria (e.g., content type, motion, spatial detail, frequency of scene cuts). Thus, we will also include HRC analyses where fewer than eight SRCs are averaged. The SRCs for these analyses will be selected to emulate unbalanced scene selection (e.g., picking all easy to code scenes).

Results for the following cases will be computed and examined:

- Per-Clip (No Common). These statistics are calculated on a per-clip basis, but all common set video sequences are discarded. The common set sequences cannot be used for the HRC analysis. Thus, the "Per-Clip (No Common)" case should be used as a baseline comparison for the HRC analyses that include scene averaging.

- 2-SRC HRC. Instead of averaging all eight SRCs associated with each HRC, the SRCs were rank-sorted by coding difficulty (i.e., determined by the average DMOS score over all coding-only HRCs), and the SRCs were paired as follows: the two easiest, the next two easiest, the next two easiest, and the two hardest. The DMOS and model scores for each HRC were averaged for these pairs of video clips (i.e., reducing the number of data points in the super-set by two, when compared to Per-Clip/No Common). Put another way, two video clips were combined to produce each synthetic "2-SRC HRC"; and for each actual HRC, there are four synthetic "2-SRC HRCs." Thus, each eight PVSs associated with one actual HRC produce four synthetic "2-SRC HRCs."

- 4-SRC HRC. Instead of averaging all eight SRCs associated with each HRC, the SRCs were rank-sorted by coding difficulty and then the SRCs were divided into two groups: the four easiest to code, and the four hardest to code. The DMOS and model scores for each HRC were averaged for each group of video clips (i.e., reducing the number of data points in the super-set by four). Thus, four video clips were combined to produce each "4-SRC HRC" data point. Put another way, four video clips were combined to produce each synthetic "4-SRC HRC"; and for each actual HRC, there are two synthetic "4-SRC HRCs." Thus, each eight PVSs associated with one actual HRC produce two synthetic "4-SRC HRCs."

- 8-SRC HRC. This is the actual HRC average computed by averaging all eight SRCs associated with each HRC (i.e., reducing the number of data points in the super-set by eight). Thus, each eight PVSs associated with one actual HRC produce one “8-SRC HRC.”

The objective and subjective data are averaged in an identical way.

The “2-SRC HRC” and “4-SRC HRC” data is indicative of HRC analysis when the video sequences available are all somewhat similar to each other in terms of coding difficulty. These analyses are included to provide an indication of accuracy improvement for HRC analysis when scenes are poorly selected. For example, the “2-SRC HRC” data could perhaps have similar performance to an instance where a user averages results from SRC that all contain similar content. Because the 1-SRC, 2-SRC, 4-SRC, and 8-SRC analyses use the same set of clips, we can track model performance as a function of averaging over an increasingly robust set of SRC.

7.1 QCIF Results

This section examines HRC video quality model performance for QCIF. The Per-Clip (No Common) performance of the objective model is compared to the 2-SRC HRC, 4-SRC HRC, and 8-SRC HRC performance.

Table 43 lists the following statistics computed using all the video sequences in the QCIF superset except the common set: Pearson correlation, RMSE, outlier ratio, and the rank groupings using RMSE. Table 44 repeats this analysis using 2-SRC HRCs (i.e., each data point represents the average of two scenes with similar coding difficulty). Table 45 and Table 46 repeat this analysis using 4-SRC and 8-SRC HRCs, respectively.

Table 43. QCIF: All Video Sequences Per-Clip (No Common)

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
PSNR	0.704	0.679	0.656						X*	
A	0.844	0.512	0.508	X*	X					
B	0.805	0.567	0.560				X	X*		
C	0.813	0.557	0.554			X	X*	X		
D	0.827	0.537	0.496		X	X*	X			
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
E	0.808	0.564	0.583				X	X*		
F	0.836	0.525	0.564	X	X*	X				
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
G	0.701	0.682	0.642						X*	
H	0.659	0.720	0.667							X*

Table 44. QCIF: HRC Analysis, All Video Sequences Averaging 2-SRC Only

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
PSNR	0.748	0.617	0.719					X	X*	X
A	0.883	0.435	0.593	X*						
B	0.837	0.506	0.652			X	X*			
C	0.844	0.496	0.651			X	X*			
D	0.853	0.481	0.596		X	X*	X			
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
E	0.847	0.494	0.661			X	X*			
F	0.868	0.460	0.617		X*	X				
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
G	0.752	0.612	0.725					X*	X	
H	0.715	0.651	0.743						X	X*

Table 45. QCIF: HRC Analysis, All Video Sequences Averaging 4-SRC Only

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.778	0.576	0.799					X*
A	0.909	0.380	0.690	X*				
B	0.854	0.467	0.725			X	X*	
C	0.863	0.454	0.730			X	X*	
D	0.875	0.434	0.670		X	X*	X	
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
E	0.872	0.447	0.728		X	X*	X	
F	0.886	0.418	0.658		X*	X		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
G	0.800	0.550	0.766					X*
H	0.791	0.588	0.824					X*

Table 46. QCIF: HRC Analysis, All Video Sequences Averaging All 8-SRC

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
PSNR	0.818	0.537	0.879							X*
A	0.919	0.335	0.723	X*						
B	0.856	0.431	0.746			X	X	X*	X	
C	0.880	0.394	0.804		X	X*	X	X		
D	0.879	0.397	0.728		X	X*	X	X		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
E	0.880	0.415	0.786		X	X	X*	X	X	
F	0.889	0.382	0.701		X*	X	X			
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**
G	0.856	0.449	0.772				X	X	X*	
H	0.810	0.533	0.871							X*

Table 47 and Figure 7 present the RMSE for each video quality model as an increasing number of scenes are averaged. The second column of Table 47, "Per-Clip (No Common)," represents the data from Table 43. The third column, "2-SRC HRC," represents the data from Table 44. The fourth column, "4-SRC HRC," represents the data from Table 45. The fifth column, "8-SRC HRC," represents the data from Table 46. The data from Table 47 is plotted in Figure 7.

Table 47 presents simplified group rankings via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for that column. FR model cells highlighted in blue identify models that are statistically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statististically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 47. QCIF: RMSE as a Function of the Number of SRCs Averaged in Each HRC

FR Models	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
PSNR	0.679	0.617	0.576	0.537
A	0.512	0.435	0.380	0.335
B	0.567	0.506	0.467	0.431
C	0.557	0.496	0.454	0.394
D	0.537	0.481	0.434	0.397
RR Model	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
E	0.564	0.494	0.447	0.415
F	0.525	0.460	0.418	0.382
NR Models	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
G	0.682	0.612	0.550	0.449
H	0.720	0.651	0.588	0.533
# of Samples	**1792**	**896**	**448**	**224**

Figure 7. QCIF: model RMSE vs. number of clips averaged. [4]

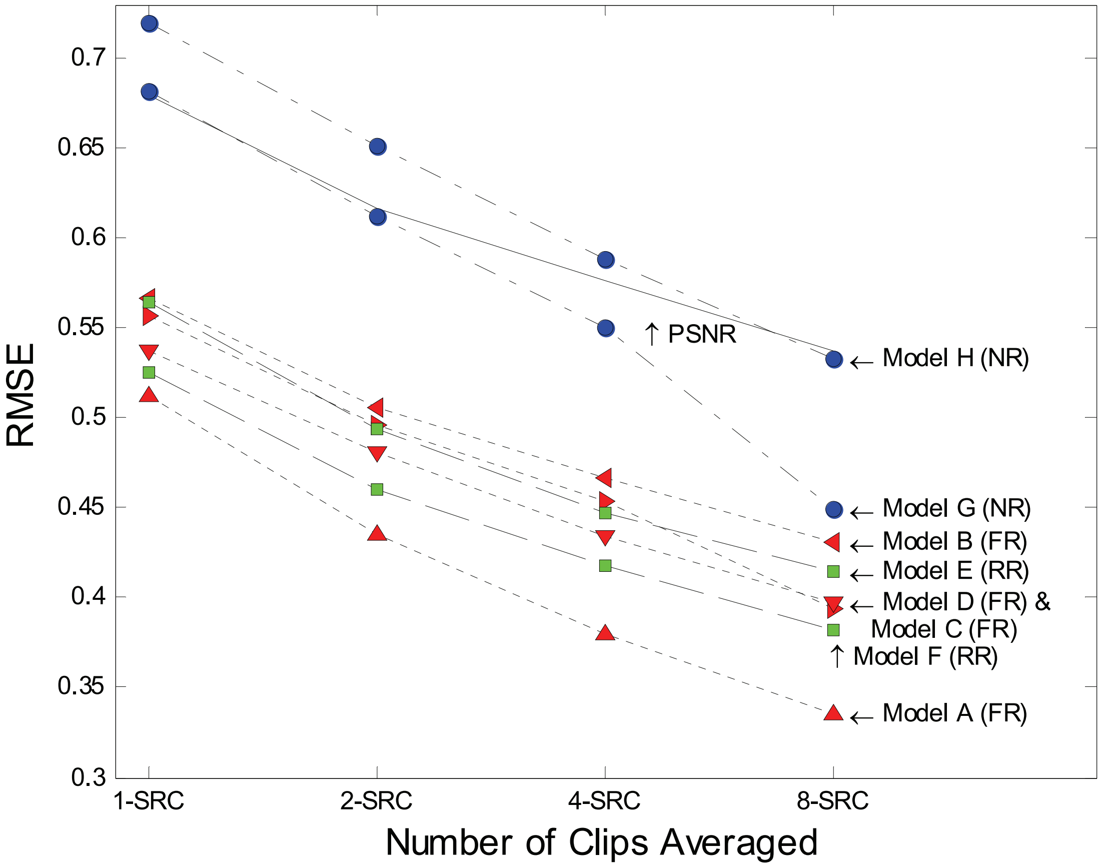
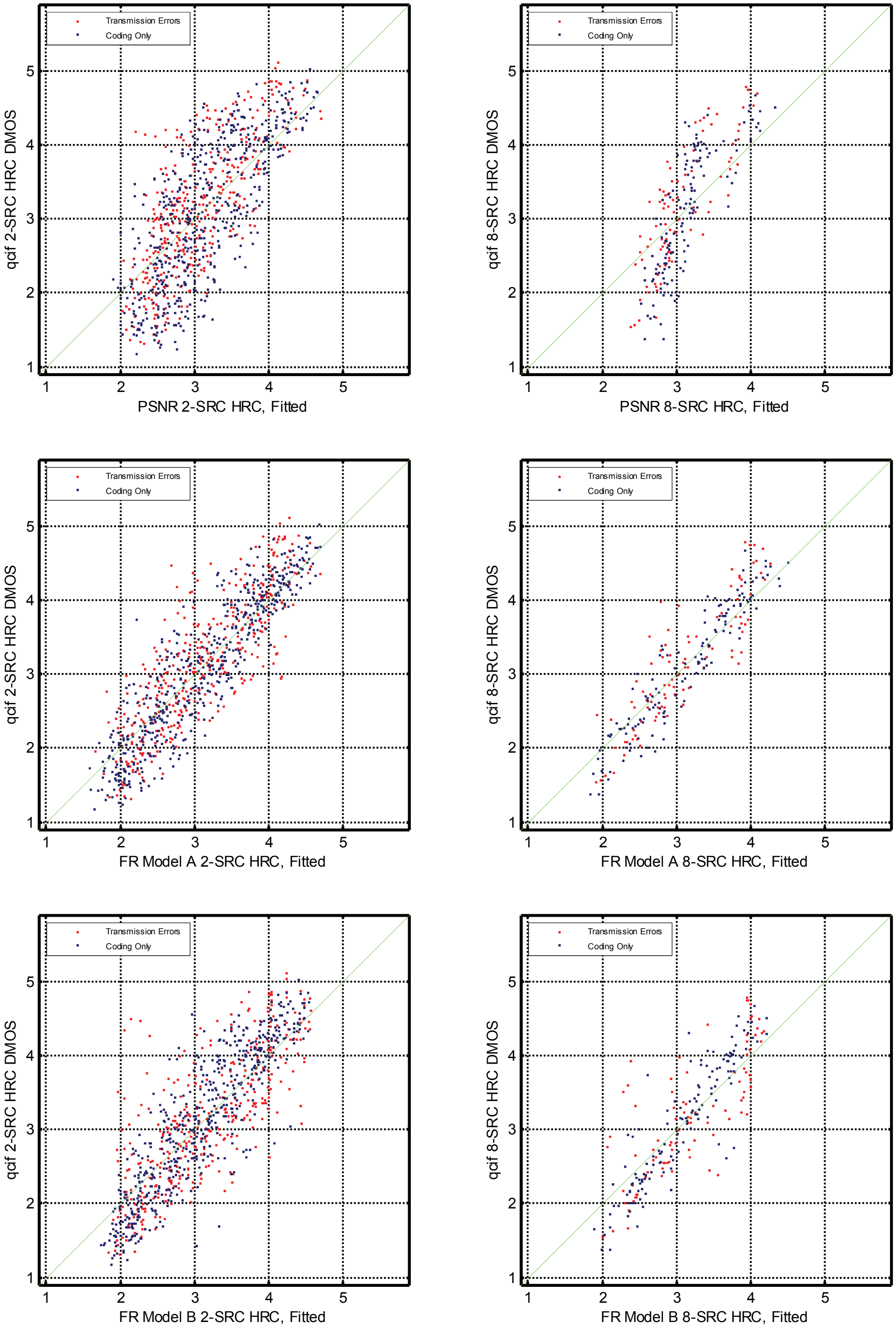


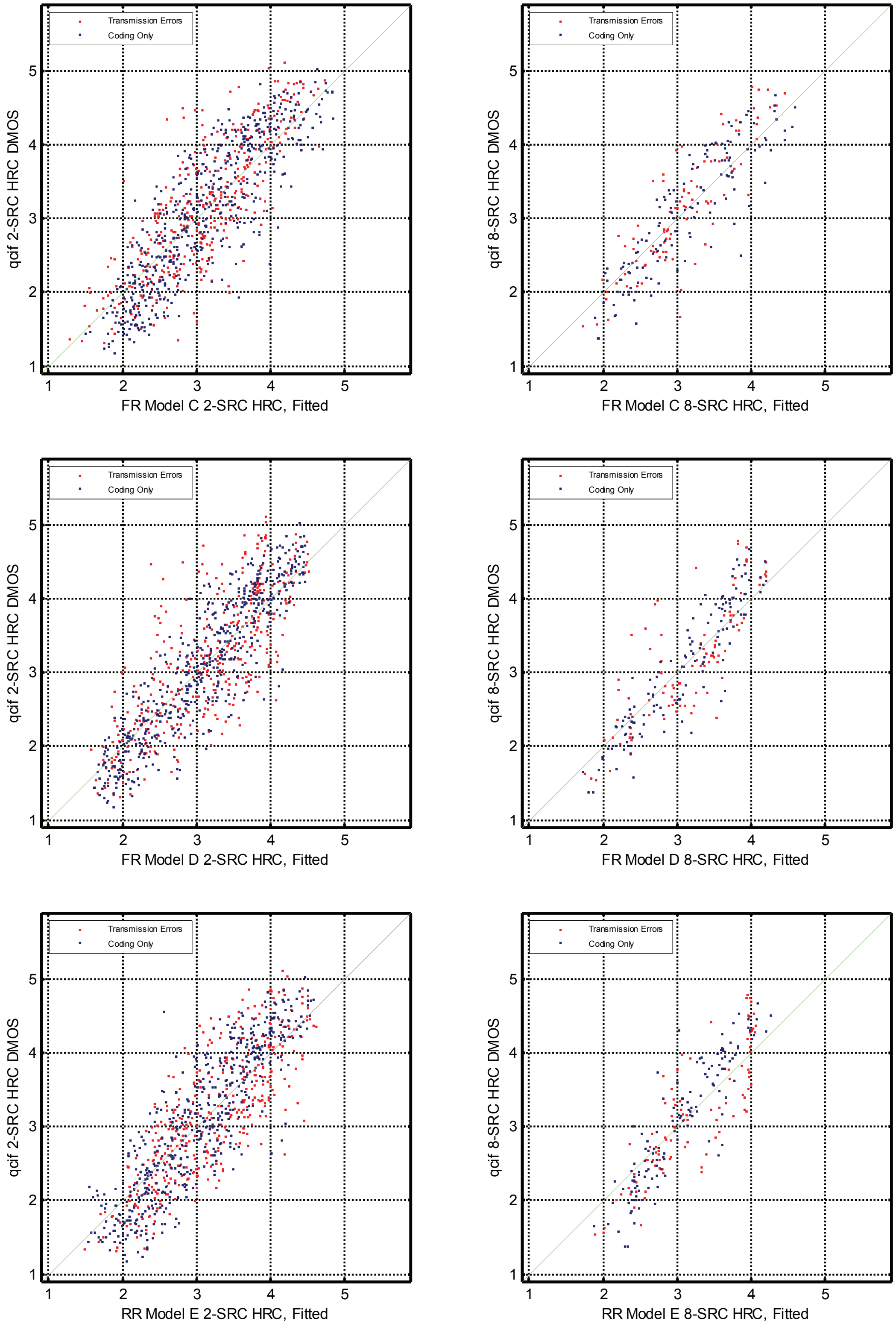
Figure 8 shows scatter plots of each model color coded to show both coding only and transmission errors, with the QCIF Superset DMOS on the y-axis and the fitted model score on the x-axis.[5] There are two adjacent plots for each model, where the left hand plot contains the 2-SRC HRC data, and the right hand plot contains the 8-SRC HRC data.

[4] Model D is plotted with a down pointing red triangle, and model C is plotted with a right pointing red triangle. PSNR is plotted with a solid line.

[5] These calculations use the same fits as previous sections (see Table 11).

Figure 8. QCIF: HRC DMOS vs. HRC model, 2-SRC HRC on left, 8-SRC HRC on right.




Transmission Errors
Coding Only
qcif 2-SRC HRC DMOS
FR Model C 2-SRC HRC, Fitted
qcif 8-SRC HRC DMOS
FR Model C 8-SRC HRC, Fitted
qcif 2-SRC HRC DMOS
FR Model D 2-SRC HRC, Fitted
qcif 8-SRC HRC DMOS
FR Model D 8-SRC HRC, Fitted
qcif 2-SRC HRC DMOS
RR Model E 2-SRC HRC, Fitted
qcif 8-SRC HRC DMOS
RR Model E 8-SRC HRC, Fitted
1
2
3
4
5

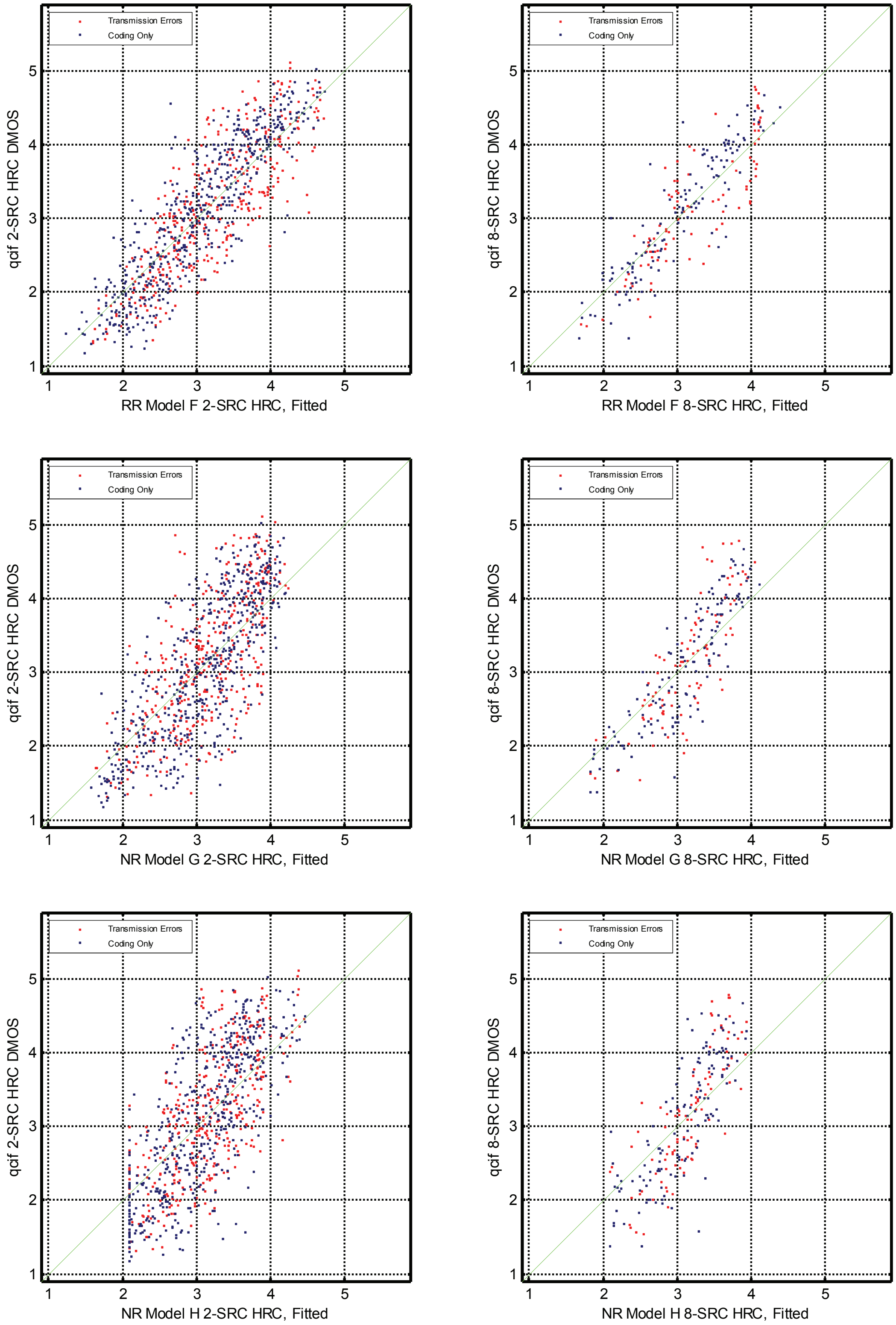


Table 48 contains the HRC resolving power (RP) for each QCIF model, computed using all 8-SRC at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power. These HRC resolving power values may be used to determine whether two HRC measurements made using a single model are significantly different. The HRC resolving power for each model on Table 48 may be compared to the clip resolving power for each model on Table 10.

Remember that direct comparisons between models should not be made using resolving power (see Section 3.3). This difference in VM range may be seen by comparing models PSNR and model C. Model C also has significantly better performance than PSNR (see Table 46) and spans a wider range of fitted VM (compare the x-axis of the 8-SRC HRC scatter plots in Figure 8 for model C and PSNR). The resolving power for model C has larger values than that of PSNR at all percentage levels due to the difference in VM range (see Table 48).

Table 48. QCIF: HRC Resolving Power

FR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
PSNR	0.67	0.52	0.24	0.16
A	0.70	0.54	0.28	0.19
B	1.04	0.75	0.29	0.19
C	0.91	0.69	0.34	0.24
D	1.05	0.61	0.29	0.20
RR Model	**95% RP**	**90% RP**	**75% RP**	**68% RP**
E	0.87	0.64	0.27	0.19
F	0.91	0.69	0.30	0.21
NR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
G	0.75	0.59	0.31	0.22
H	0.76	0.58	0.29	0.20

Our interpretation of the QCIF results is as follows:

- HRC processing appears to improve the models' RMSE accuracy (see Figure 7 and Table 47). Such averaging may or may not be appropriate depending upon the application of the objective video quality model.

- RMSE accuracy improves steadily for all models as the number of scenes used increases from 1-SRC to 2-SRC to 4-SRC to 8-SRC. This indicates that while scene selection is important for estimating HRC quality, some improvement can probably be obtained even when scenes are poorly chosen.

- The rate at which a model's RMSE drops as the number of scenes averaged increases is similar for all models (see Figure 7).

- When measuring average HRC quality, the FR model A appears to be statistically better than all other models.

- When measuring average HRC quality, the RR models E & F and NR models G & H appear to be statistically equivalent to or better than PSNR.

- Several models have statistically equivalent performance for measuring average HRC quality but statistically different performance for measuring per-clip quality (e.g., compare FR model C per-clip performance in Table 43 with the 8-SRC HRC performance in Table 46).

7.2 CIF Results

This section examines HRC video quality model performance for CIF. The Per-Clip (No Common) performance of the objective model is compared to the 2-SRC HRC, 4-SRC HRC, and 8-SRC HRC performance.

Table 49 lists the following statistics computed using all the video sequences in the CIF superset except the common set: Pearson correlation, RMSE, outlier ratio, and the ranking groups using RMSE. Table 50 repeats this analysis using 2-SRC HRCs (i.e., each data point represents the average of two scenes with similar coding difficulty). Table 51 and Table 52 repeat this analysis using 4-SRC and 8-SRC HRCs, respectively.

Table 49. CIF: All Video Sequences Per-Clip (No Common)

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.644	0.732	0.687				X*	
I	0.796	0.580	0.568		X*			
J	0.760	0.622	0.595			X*		
K	0.848	0.507	0.542	X*				
L	0.796	0.580	0.586		X*			
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.775	0.606	0.599			X*		
N	0.775	0.605	0.593			X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.482	0.839	0.718					X*
P	0.520	0.818	0.700					X*

Table 50. CIF: HRC Analysis, All Video Sequences Averaging 2-SRC Only

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.709	0.657	0.773				X*	
I	0.853	0.484	0.657		X*			
J	0.799	0.553	0.703			X*		
K	0.893	0.415	0.612	X*				
L	0.853	0.489	0.693		X*			
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.816	0.532	0.674			X*		
N	0.817	0.531	0.680			X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.550	0.776	0.796					X*
P	0.591	0.744	0.756					X*

Table 51. CIF: HRC Analysis, All Video Sequences Averaging 4-SRC Only

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5
PSNR	0.725	0.621	0.813				X*	
I	0.874	0.436	0.743		X*			
J	0.814	0.509	0.772			X*		
K	0.908	0.370	0.723	X*				
L	0.873	0.446	0.766		X*			
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.829	0.492	0.754			X*		
N	0.829	0.492	0.759			X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.616	0.724	0.826					X*
P	0.651	0.677	0.797					X*

Table 52. CIF: HRC Analysis, All Video Sequences Averaging All 8-SRC

FR Models	Correlation	RMSE	OR	G1	G2	G3	G4	G5
PSNR	0.685	0.595	0.853					X*
I	0.872	0.400	0.817		X*	X		
J	0.809	0.459	0.804			X	X*	
K	0.915	0.324	0.746	X*				
L	0.870	0.414	0.862		X	X*	X	
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
M	0.810	0.460	0.835			X	X*	
N	0.811	0.460	0.839			X	X*	
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
O	0.631	0.640	0.862					X*
P	0.699	0.575	0.821					X*

Table 53 and Figure 9 present the RMSE for each video quality model as an increasing number of scenes are averaged. The second column of Table 53, “Per-Clip (No Common),” represents the data from Table 49. The third column, “2-SRC HRC,” represents the data from Table 50. The fourth column, “4-SRC HRC,” represents the data from Table 51. The fifth column, “8-SRC HRC,” represents the data from Table 52. The data from Table 53 is plotted in Figure 9.

Table 53 presents simplified group rankings via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for that column. FR model cells highlighted in blue identify models that are statististically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statistically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 53. CIF: RMSE as a Function of the Number of SRCs Averaged in Each HRC

FR Models	Per-Clip (No Common)	2-SRC HRC	4-SRC HRC	8-SRC HRC
PSNR	0.732	0.657	0.621	0.595
I	0.580	0.484	0.436	0.400
J	0.622	0.553	0.509	0.459
K	0.507	0.415	0.370	0.324
L	0.580	0.489	0.446	0.414
RR Model	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
M	0.606	0.532	0.492	0.460
N	0.605	0.531	0.492	0.460
NR Models	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
O	0.839	0.776	0.724	0.640
P	0.818	0.744	0.677	0.575
# of Samples	**1792**	**896**	**448**	**224**

Figure 9. CIF: model RMSE vs. number of clips averaged.

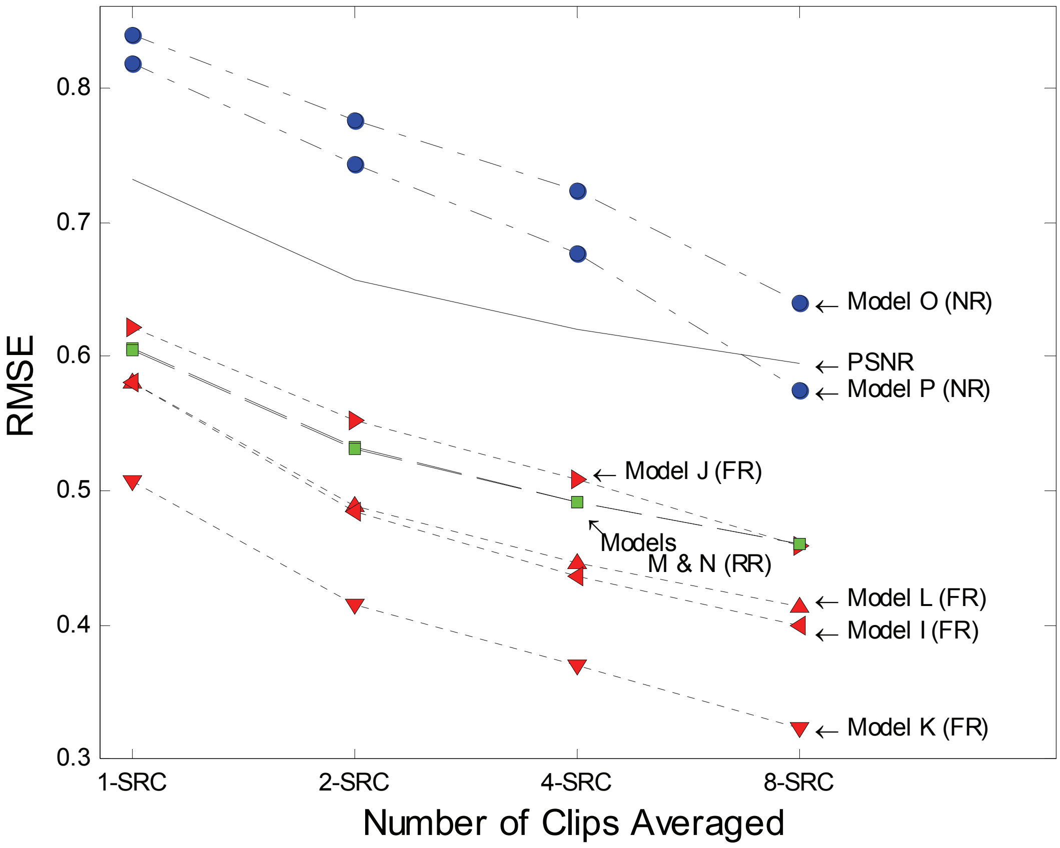
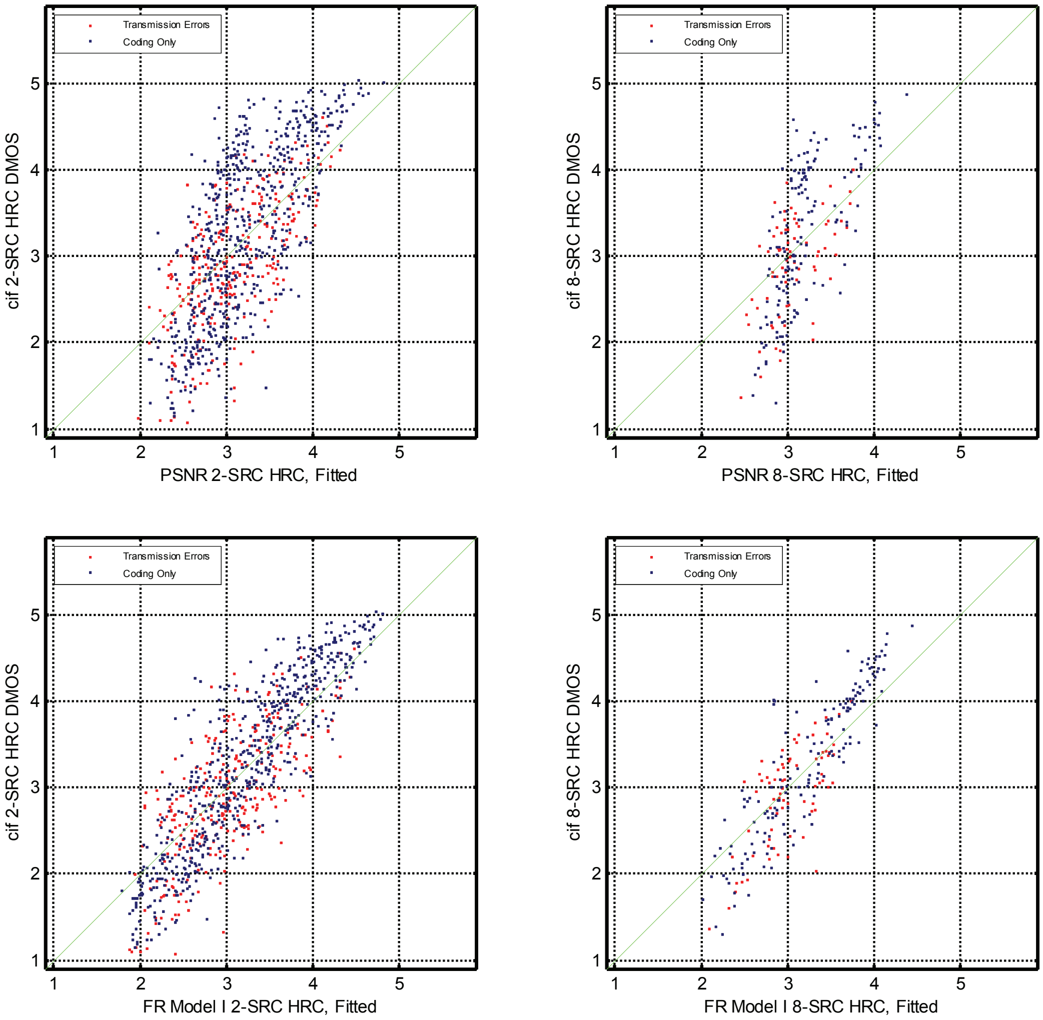
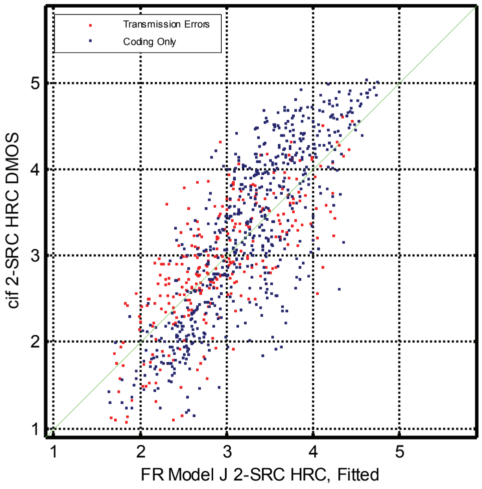


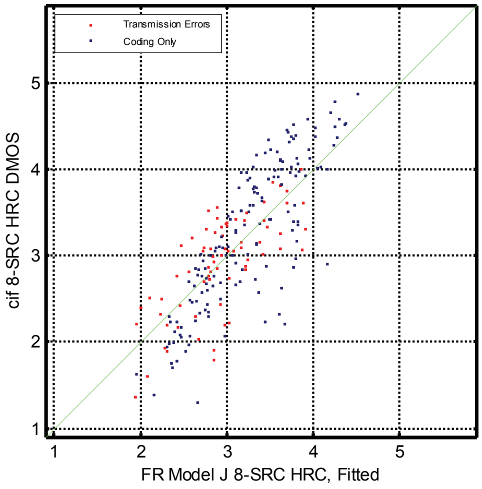
Figure 10 shows scatter plots of each model color coded to show both coding only and transmission errors, with the CIF Superset DMOS on the y-axis and the fitted model score on the x-axis.[6] There are two adjacent plots for each model, where the left hand plot contains the 2-SRC HRC data, and the right hand plot contains the 8-SRC HRC data.

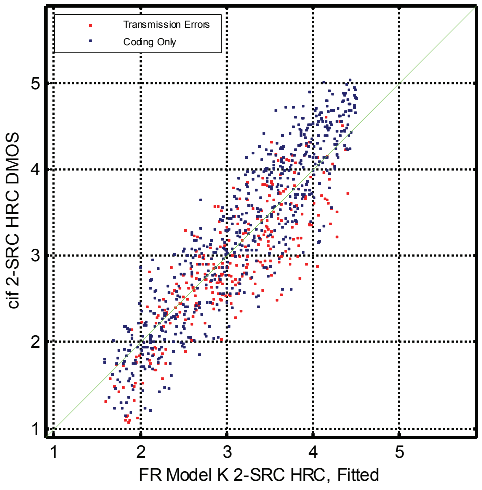
Figure 10. CIF: HRC DMOS vs. HRC model, 2-SRC HRC on left, 8-SRC HRC on right.

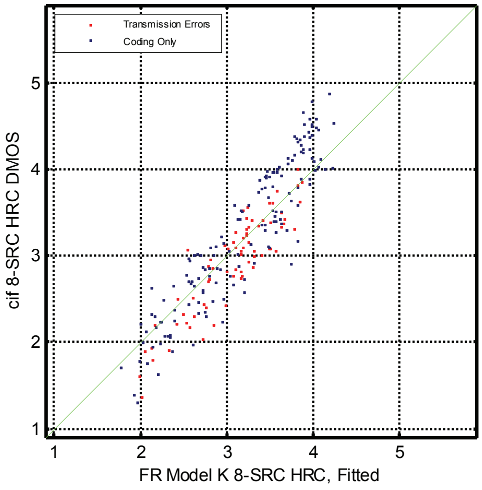


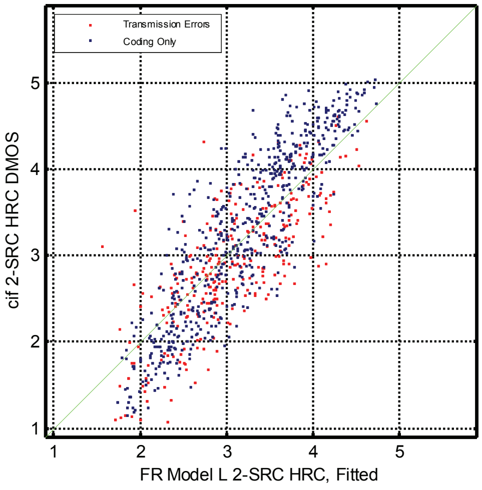
[6] These calculations use the same fits as previous sections (see Table 16).

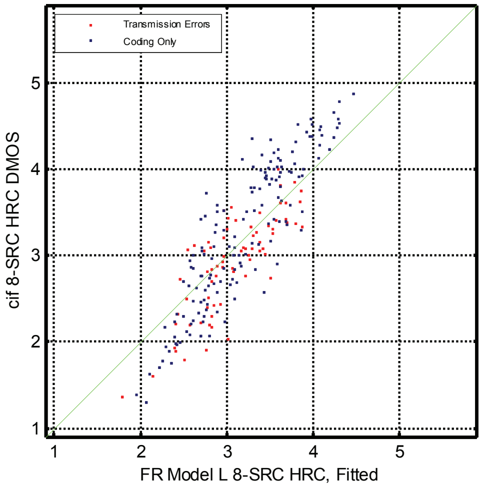

Transmission Errors
Coding Only
cif 2-SRC HRC DMOS
FR Model J 2-SRC HRC, Fitted
1
2
3
4
5

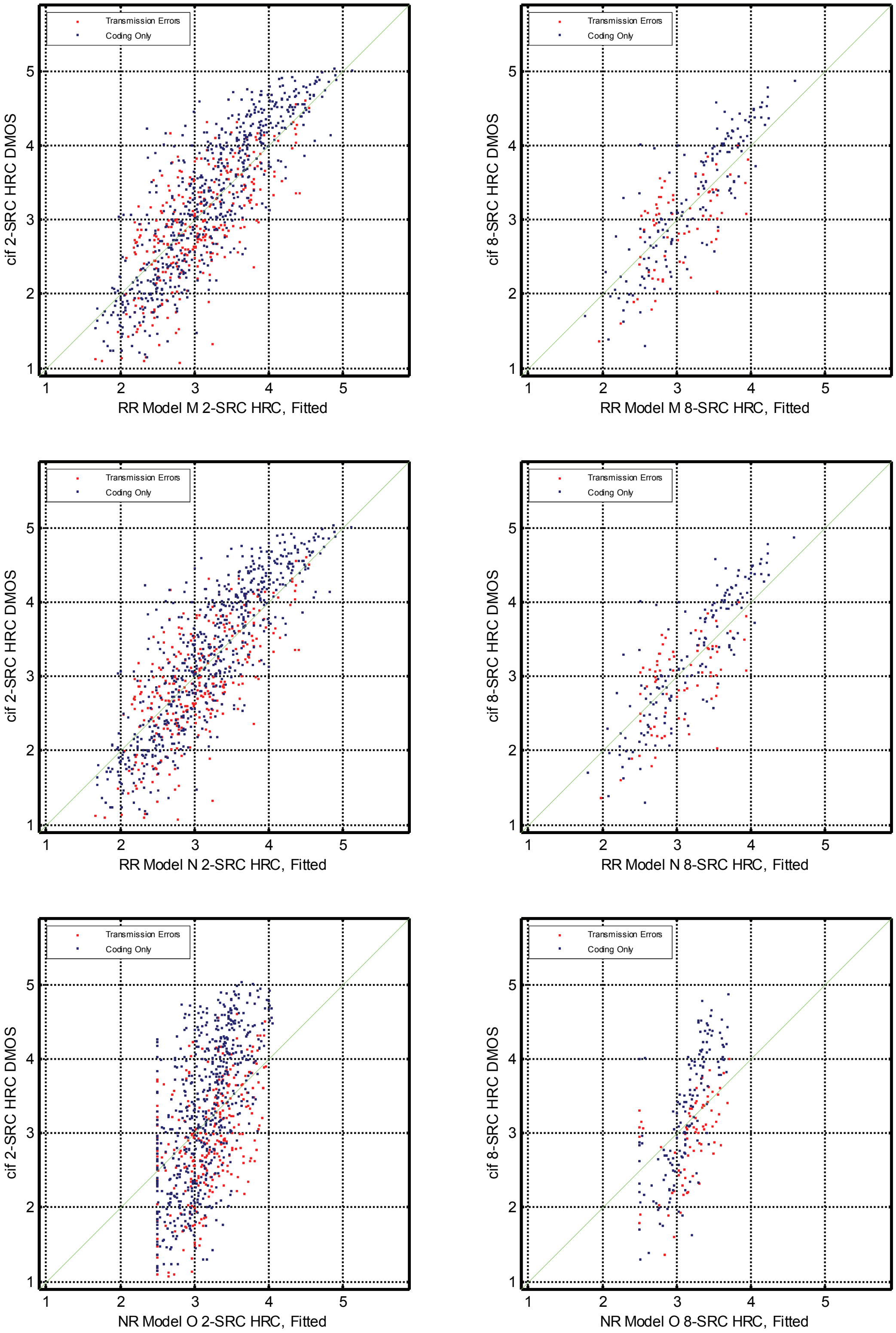

Transmission Errors
Coding Only
cif 8-SRC HRC DMOS
FR Model J 8-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
cif 2-SRC HRC DMOS
FR Model K 2-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
cif 8-SRC HRC DMOS
FR Model K 8-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
cif 2-SRC HRC DMOS
FR Model L 2-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
cif 8-SRC HRC DMOS
FR Model L 8-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
cif 2-SRC HRC DMOS
RR Model M 2-SRC HRC, Fitted
cif 8-SRC HRC DMOS
RR Model M 8-SRC HRC, Fitted
cif 2-SRC HRC DMOS
RR Model N 2-SRC HRC, Fitted
cif 8-SRC HRC DMOS
RR Model N 8-SRC HRC, Fitted
cif 2-SRC HRC DMOS
NR Model O 2-SRC HRC, Fitted
cif 8-SRC HRC DMOS
NR Model O 8-SRC HRC, Fitted
1
2
3
4
5

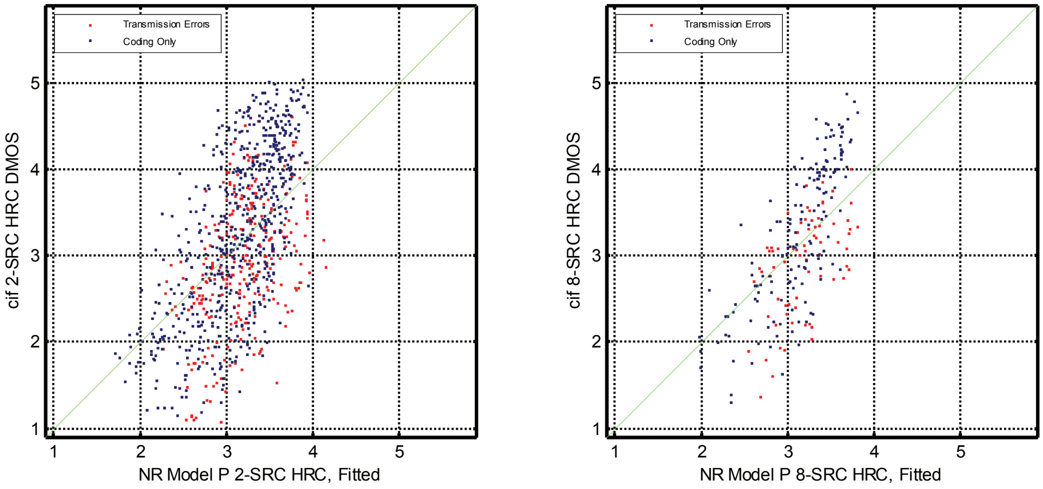


Table 54 contains the HRC resolving power (RP) for each CIF model, computed using all 8-SRC at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power. These HRC resolving power values may be used to determine whether two HRC measurements made using a single model are significantly different. The HRC resolving power for each model in Table 54 may be compared to the clip resolving power for each model in Table 15.

Remember that direct comparisons between models should not be made using resolving power (see Section 3.3). This difference in VM range may be seen by comparing models PSNR and model J. Model J also has significantly better performance than PSNR (see Table 52) and spans a wider range of fitted VM (compare the x-axis of the 8-SRC HRC scatter plots in Figure 10 for model J and PSNR). The resolving power for model J has larger values than that of PSNR at all percentage levels due to the difference in VM range (see Table 54).

Table 54. CIF: HRC Resolving Power

FR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
PSNR	0.84	0.74	0.28	0.17
I	0.84	0.63	0.30	0.19
J	1.08	0.83	0.38	0.25
K	0.69	0.53	0.27	0.19
L	0.75	0.57	0.32	0.22
RR Model	**95% RP**	**90% RP**	**75% RP**	**68% RP**
M	1.08	0.82	0.39	0.27
N	1.07	0.82	0.38	0.26
NR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
O	1.11	1.02	0.23	0.15
P	1.02	0.74	0.35	0.23

Our interpretation of the CIF results is presented here.

- HRC processing appears to improve the models' RMSE accuracy (see Figure 9 and Table 53). Such averaging may or may not be appropriate depending upon the application of the objective video quality model.

- RMSE accuracy improves steadily for all models as the number of scenes used increases from 1-SRC to 2-SRC to 4-SRC to 8-SRC. This indicates that while scene selection is important for estimating HRC quality, some improvement can probably be obtained even when scenes are poorly chosen.

- The rate at which a model's RMSE drops as the number of scenes averaged increases is similar for all models (see Figure 9).

- When measuring average HRC quality, the FR model K appears to be statistically better than all other models.

- When measuring average HRC quality, the RR models M & N appear to be statistically equivalent to or better than PSNR (see Table 53).

7.3 VGA Results

This section examines HRC video quality model performance for VGA. The Per-Clip (No Common) performance of the objective model on a per-clip basis is compared to the 2-SRC HRC, 4-SRC HRC, and 8-SRC HRC performance.

Table 55 lists the following statistics computed using all the video sequences in the VGA superset except the common set: Pearson correlation, RMSE, outlier ratio, and the ranking groups using RMSE. Table 56, Table 57, and Table 58 repeat this analysis using 2-SRC HRCs, 4-SRC HRCs, and 8-SRC HRCs, respectively.

Table 55. VGA: All Video Sequences Per-Clip (No Common)

FR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
PSNR	0.727	0.699	0.663				X*	
Q	0.817	0.586	0.594	X*	X			
R	0.738	0.686	0.657				X*	
S	0.802	0.608	0.593	X	X*	X		
T	0.795	0.618	0.599		X	X*		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
U	0.798	0.614	0.613		X	X*		
V	0.798	0.613	0.611		X	X*		
W	0.799	0.613	0.613		X	X*		
NR Models	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**
X	0.417	0.925	0.771					X*
Y	0.433	0.918	0.741					X*

Table 56. VGA: HRC Analysis, All Video Sequences Averaging 2-SRC Only

FR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
PSNR	0.787	0.608	0.746			X*	
Q	0.872	0.486	0.661	X*			
R	0.787	0.605	0.726			X*	
S	0.838	0.534	0.667		X*		
T	0.842	0.530	0.671		X*		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
U	0.841	0.531	0.673		X*		
V	0.841	0.530	0.672		X*		
W	0.843	0.527	0.673		X*		
NR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
X	0.451	0.885	0.837				X*
Y	0.458	0.870	0.806				X*

Table 57. VGA: HRC Analysis, All Video Sequences Averaging 4-SRC Only

FR Model	Correlation	RMSE	OR	G1	G2	G3	G4
PSNR	0.799	0.565	0.785			X*	
Q	0.898	0.423	0.727	X*			
R	0.824	0.531	0.746			X*	
S	0.861	0.474	0.766		X*		
T	0.857	0.481	0.744		X*		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
U	0.856	0.481	0.727		X*		
V	0.856	0.481	0.729		X*		
W	0.858	0.478	0.722		X*		
NR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**
X	0.482	0.835	0.868				X*
Y	0.504	0.808	0.890				X*

Table 58. VGA: HRC Analysis, All Video Sequences Averaging All 8-SRC

FR Model	Correlation	RMSE	OR	G1	G2	G3	G4	G5	G6
PSNR	0.792	0.515	0.868				X	X*	
Q	0.906	0.373	0.722	X*					
R	0.816	0.480	0.800			X	X*	X	
S	0.859	0.425	0.766		X*	X			
T	0.859	0.429	0.805		X	X*	X		
RR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
U	0.857	0.426	0.732		X*	X			
V	0.857	0.426	0.737		X*	X			
W	0.860	0.423	0.741		X*	X			
NR Model	**Correlation**	**RMSE**	**OR**	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
X	0.502	0.741	0.917						X*
Y	0.485	0.730	0.829						X*

Table 59 and Figure 11 present the RMSE for each video quality model as an increasing number of scenes are averaged. The second column of Table 59, "Per-Clip (No Common)," represents the data from Table 55. The third column, "2-SRC HRC," represents the data from Table 56. The fourth column, "4-SRC HRC," represents the data from Table 57. The fifth column, "8-SRC HRC," represents the data from Table 58. The data from Table 59 is plotted in Figure 11.

Table 59 presents simplified group rankings via colored highlights. Cells highlighted in yellow identify models that are statistically equivalent at the 95% significance level (using the F-test) to the top performing model for that column. FR model cells highlighted in blue identify models that are statistically better than PSNR yet statistically worse than the top performing model. RR and NR model cells highlighted in turquoise identify models that are statistically equivalent to or better than PSNR yet statistically worse than the top performing model (less stringent criteria are specified for RR and NR because PSNR cannot be used in these environments).

Table 59. VGA: RMSE as a Function of the Number of SRC Averaged in Each HRC

FR Models	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
PSNR	0.699	0.608	0.565	0.515
Q	0.586	0.486	0.423	0.373
R	0.686	0.605	0.531	0.480
S	0.608	0.534	0.474	0.425
T	0.618	0.530	0.481	0.429
RR Model	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
U	0.614	0.531	0.481	0.426
V	0.613	0.530	0.481	0.426
W	0.613	0.527	0.478	0.423
NR Models	**Per-Clip (No Common)**	**2-SRC HRC**	**4-SRC HRC**	**8-SRC HRC**
X	0.925	0.885	0.835	0.741
Y	0.918	0.870	0.808	0.730
# of Samples	**1640**	**820**	**410**	**205**

Figure 11. VGA: model RMSE vs. number of clips averaged.

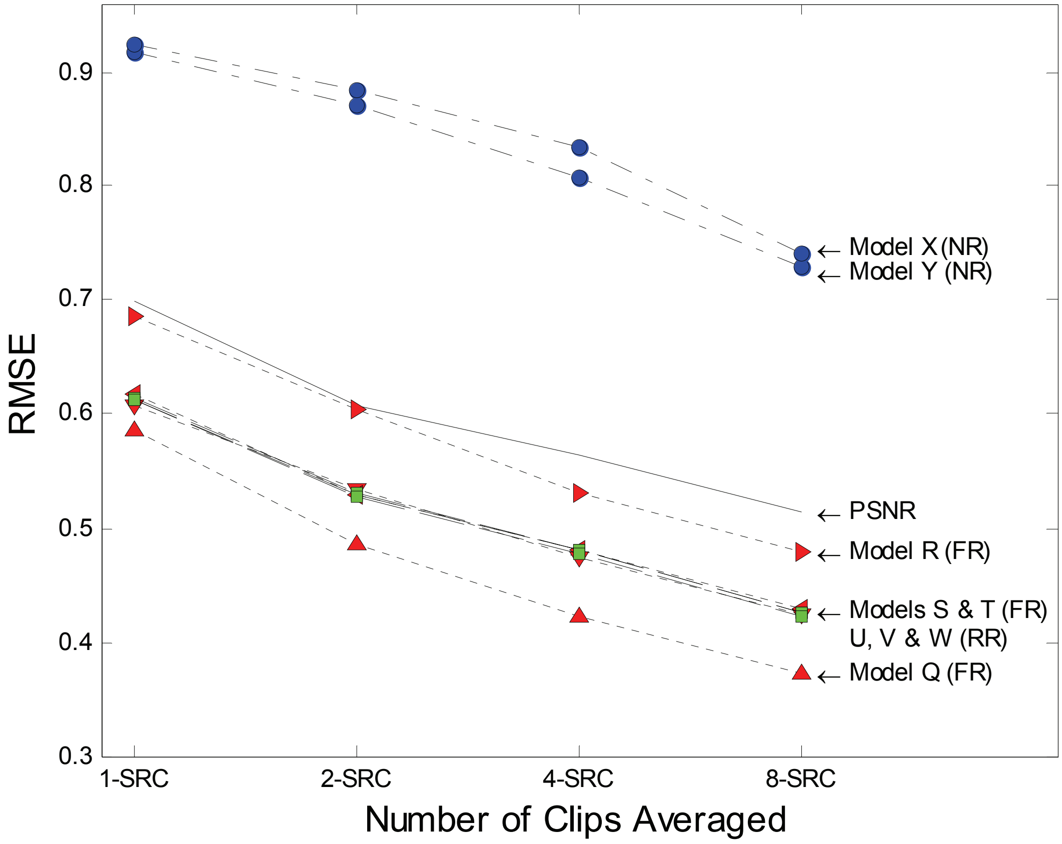
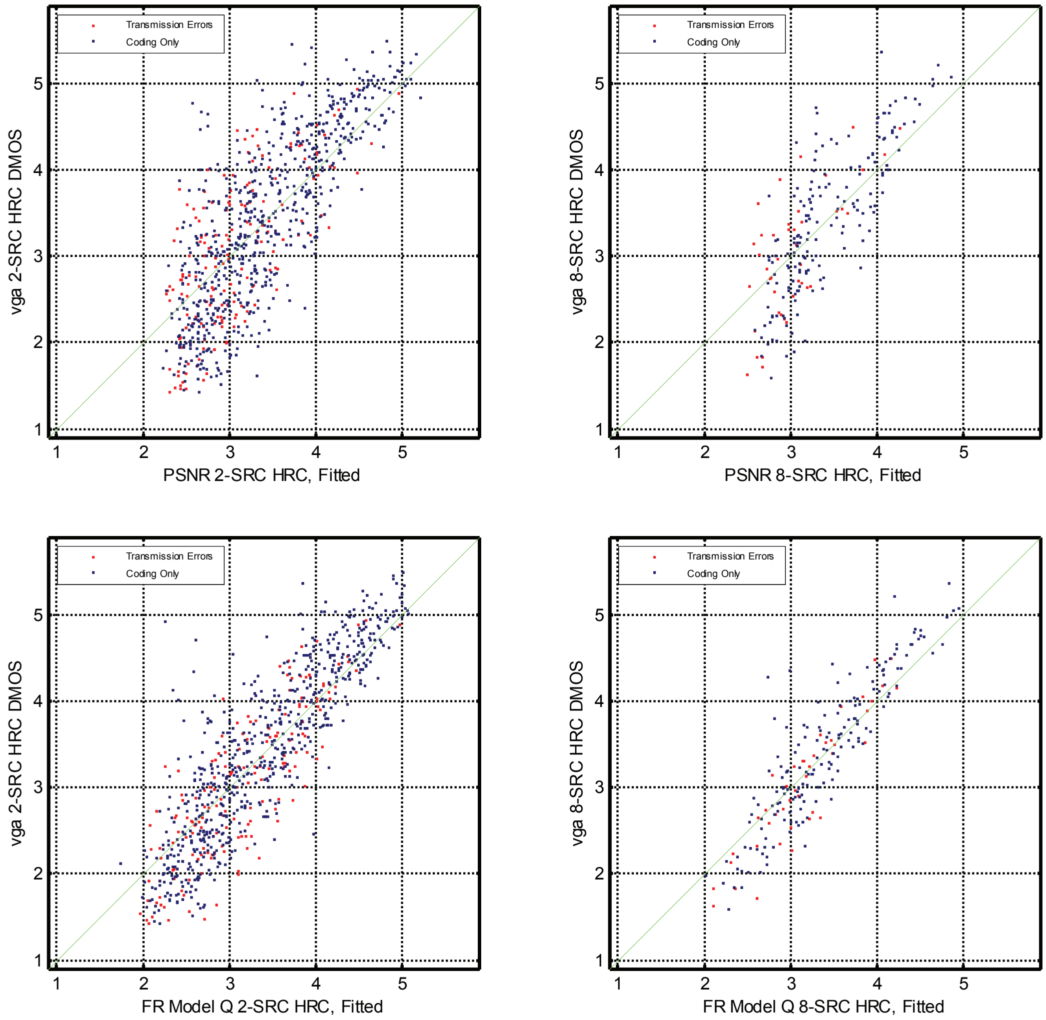
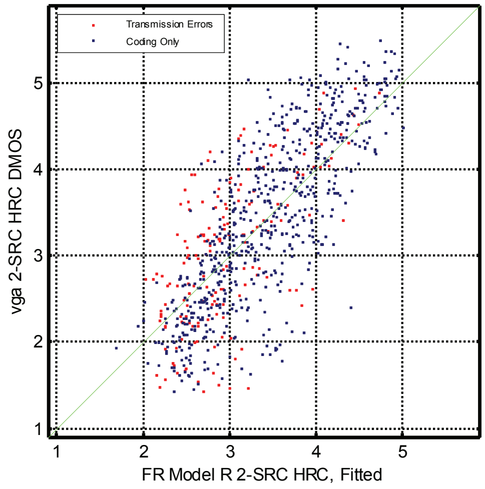


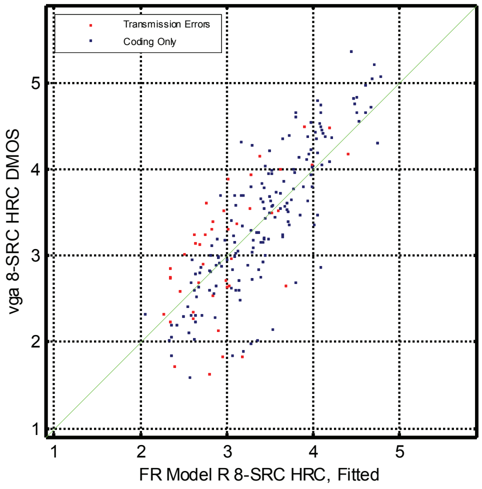
Figure 12 shows scatter plots of each model color coded to show both coding only and transmission errors, with the VGA Superset DMOS on the y-axis and the fitted model score on the x-axis.[7] There are two adjacent plots for each model, where the left hand plot contains the 2-SRC HRC data, and the right hand plot contains the 8-SRC HRC data.

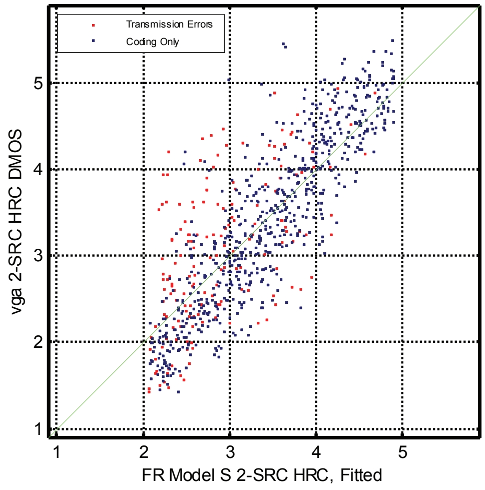
Figure 12. VGA: HRC DMOS vs. HRC model, 2-SRC HRC on left, 8-SRC HRC on right.

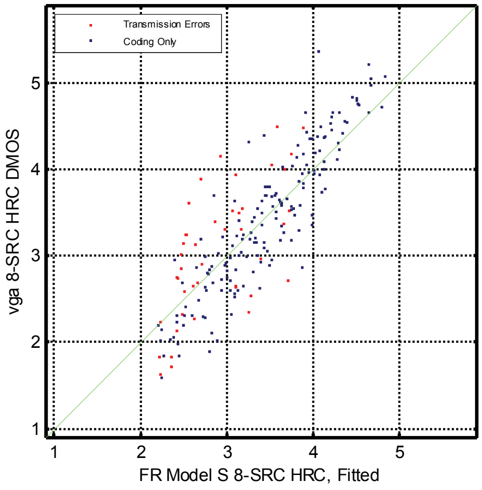


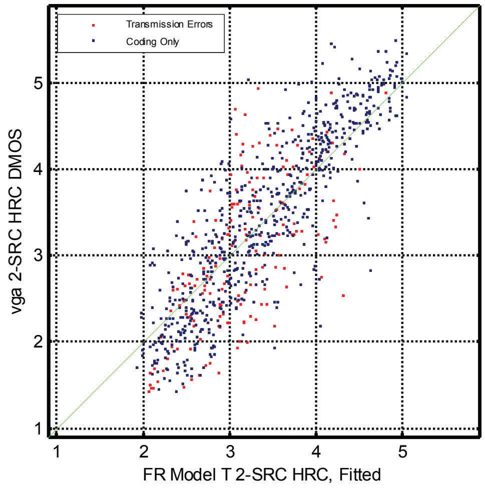
[7] These calculations use the same fits as previous sections (see Table 21).

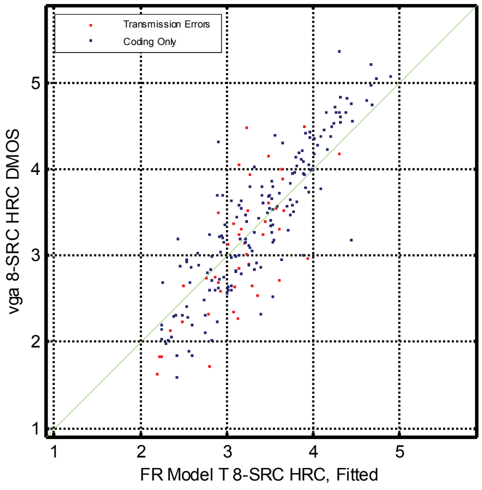

Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
FR Model R 2-SRC HRC, Fitted
1
2
3
4
5

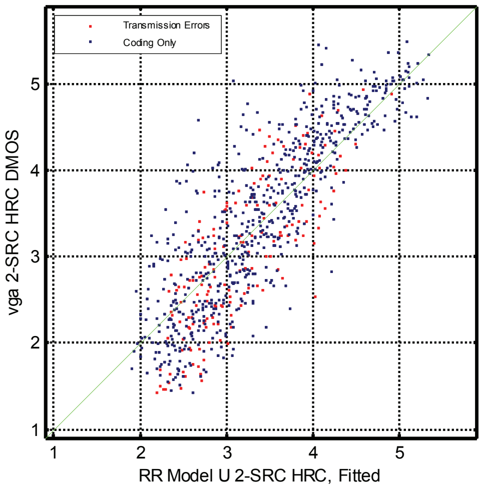

Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
FR Model R 8-SRC HRC, Fitted
1
2
3
4
5

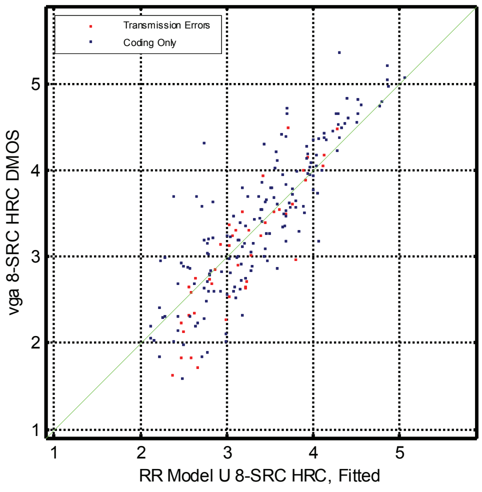

Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
FR Model S 2-SRC HRC, Fitted
1
2
3
4
5

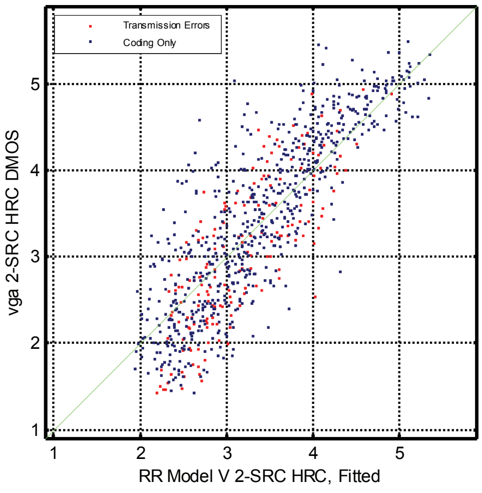

Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
FR Model S 8-SRC HRC, Fitted
1
2
3
4
5

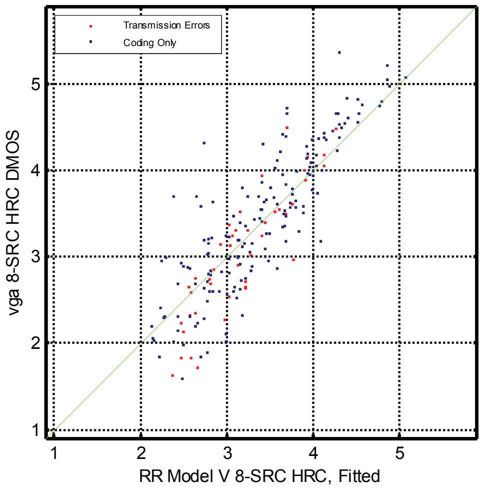

Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
FR Model T 2-SRC HRC, Fitted
1
2
3
4
5

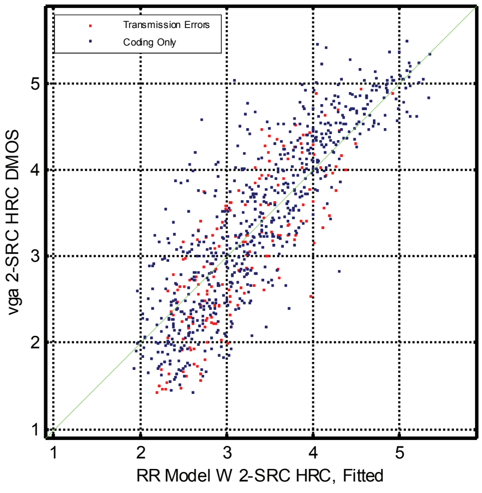

Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
FR Model T 8-SRC HRC, Fitted
1
2
3
4
5

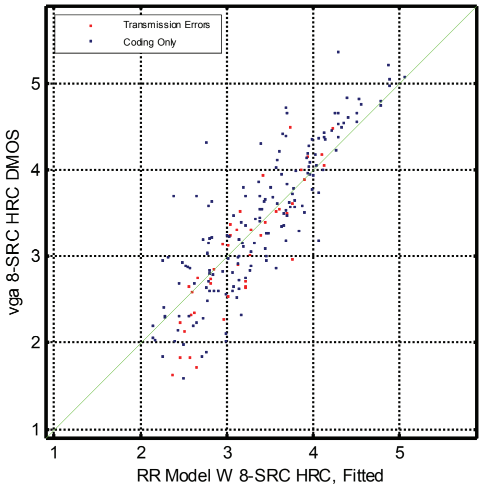

Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
RR Model U 2-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
RR Model U 8-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
RR Model V 2-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
RR Model V 8-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
vga 2-SRC HRC DMOS
RR Model W 2-SRC HRC, Fitted
1
2
3
4
5


Transmission Errors
Coding Only
vga 8-SRC HRC DMOS
RR Model W 8-SRC HRC, Fitted
1
2
3
4
5

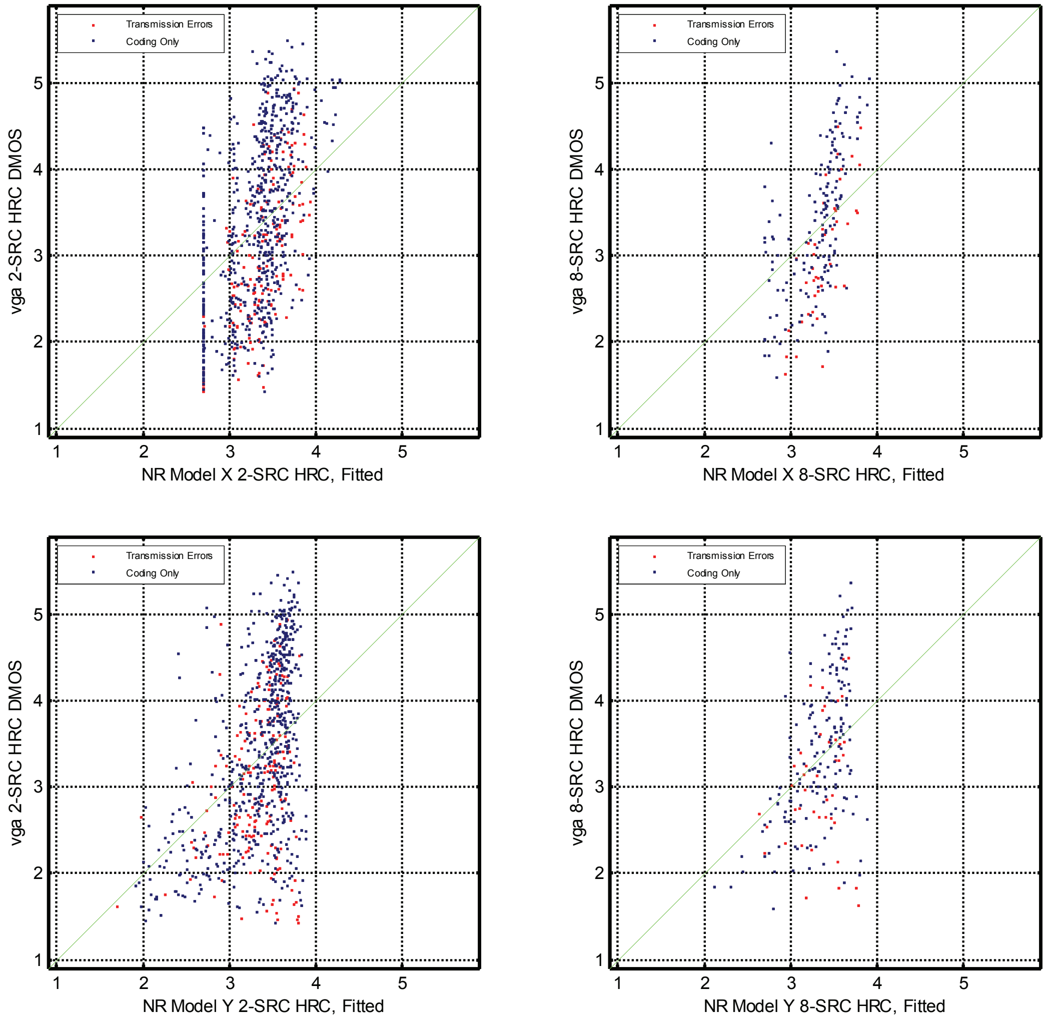


Table 60 contains the HRC resolving power for each VGA model, computed using all 8-SRC at four confidence levels: 95% resolving power, 90% resolving power, 75% resolving power, and 68% resolving power. These HRC resolving power values may be used to determine whether two HRC measurements made using a single model are significantly different. The HRC resolving power for each model in Table 60 may be compared to the clip resolving power for each model in Table 20.

Remember that direct comparisons between models should not be made using resolving power (see Section 3.3). This difference in VM range may be seen by comparing models PSNR and model W. RR model W also has significantly better performance than PSNR (see Table 58) and spans a wider range of fitted VM (compare the x-axis of the 8-SRC HRC scatter plots in Figure 12 for model W and PSNR). However, the resolving power for model W has nearly identical values to those of PSNR at all percentage levels (see Table 60).

Table 60. VGA: HRC Resolving Power

FR Models	95% RP	90% RP	75% RP	68% RP
PSNR	0.99	0.84	0.36	0.22
Q	0.68	0.52	0.27	0.19
R	1.04	0.83	0.42	0.29
S	1.04	0.80	0.38	0.25
T	0.99	0.73	0.35	0.24
RR Model	**95% RP**	**90% RP**	**75% RP**	**68% RP**
U	1.01	0.73	0.39	0.26
V	1.01	0.73	0.39	0.26
W	0.99	0.72	0.38	0.26
NR Models	**95% RP**	**90% RP**	**75% RP**	**68% RP**
X	0.88	0.80	0.66	0.06
Y	1.40	1.08	0.46	0.30

Our interpretation of the VGA results is presented here.

- HRC processing appears to improve the models' RMSE accuracy (see Figure 11 and Table 59). Such averaging may or may not be appropriate depending upon the application of the objective video quality model.

- RMSE accuracy improves steadily for all models as the number of scenes used increases from 1-SRC to 2-SRC to 4-SRC to 8-SRC. This indicates that while scene selection is important for estimating HRC quality, some improvement can probably be obtained even when scenes are poorly chosen.

- The rate at which a model's RMSE drops as the number of scenes averaged increases is similar for all models (see Figure 11).

- When measuring average HRC quality, the FR model Q appears to be statistically better than all other models.

- When measuring average HRC quality, the RR models U, V, and W appear to be statistically equivalent to or better than PSNR (see Table 59).

8 CONCLUSIONS

We have presented a methodology for combining many subjective tests from multiple laboratories into a large superset of subjective scores. This methodology utilized a common set of video sequences that were included in all the subjective experiments and that spanned the full range of subjective quality that was present in the experiments. The genesis for the methodology was the excellent laboratory-to-laboratory cross correlations of a common set of video sequences that were included in the VQEG MM Phase I experiments at all three image resolutions (QCIF, CIF, and VGA).

The subjective data superset analysis presented in this document provides a level of insight into the relative performance of the VQEG MM Phase I models, and also into their expected performance when applied by end-users. The results from this supplemental analysis enable more powerful and useful conclusions to be reached than if each experiment is examined separately. With the combined superset analysis, we were able to examine the performance of the objective models for coding only errors versus coding plus transmission errors, for particular types of codecs, and for estimating average HRC quality. In addition, the superset analysis enables the computation of resolving power which provides end-users with a quantitative method for computing the accuracy of their objective metrics. Since the superset spans many more scenes and video systems, we believe these results better represent the relative and overall accuracies achieved by each model.

The supserset analysis also provides an additional level of insight into the relative performance for each type of model. For each resolution (QCIF, CIF, and VGA), this document provides analysis on four FR models, and two NR models. The performance of the various model types was examined for coding only impairments (e.g., MPEG-4, H.264, VC-1, RV-10) and coding plus transmission errors. For each image resolution, a top performing model can be found with consistent performance between the coding only and coding plus transmission errors categories. This top performing model is always a FR model. Also, a RR model can be found at each video resolution that is often in the top performing group and always at least as accurate as FR model PSNR (in an RMSE sense). NR models are seldom in the top performing group, yet occasionally achieve the accuracy of PSNR.

From this analysis, we can also provide better guidelines to help design future validation testing. First, common set video sequences worked well and should be considered for all future tests (when multiple laboratories are performing the testing). Second, future validation tests that utilize multiple experiments should consider specifying an approximate distribution of major factors (e.g., codec type, transmission errors), to avoid “holes” in the resulting data. This will also better distribute HRCs across the variables of interest to standards committees (e.g., specific coding algorithms to be mentioned in the scope of a standard).

9 ACKNOWLEDGEMENTS

The analysis presented in this paper was made possible through the efforts of the many organizations that contributed to the VQEG MM Phase I experiment. Particular thanks go to Greg Cermak of Verizon for double-checking key statistical assumptions and calculations.

10 REFERENCES

[1] "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Multimedia Quality Assessment, Phase I," available at http://www.its.bldrdoc.gov/vqeg/projects/multimedia/.

[2] ATIS T1.TR.72-2003 "Methodological Framework for Specifying Accuracy and Cross-Calibration of Video Quality Metrics," available at https://www.atis.org/docstore/product.aspx?id=10518.

[3] ITU-T Recommendation J.149 (03/04), "Method for specifying accuracy and cross-calibration of Video Quality Metrics (VQM)," available at http://www.itu.int/rec/T-REC-J.149/en.

[4] M.H. Brill, J. Lubin, P. Costa, S. Wolf, and J. Pearson, "Accuracy and cross-calibration of video quality metrics: new methods from ATIS/T1A1," *Signal Processing: Image Communication*, Vol. 19, Feb. 2004, pp. 101-107.

APPENDIX A: PEAK SIGNAL-TO-NOISE RATIO (PSNR)

The Peak Signal to Noise Ratio (PSNR) calculation used an exhaustive search method for computing PSNR. This algorithm performed an exhaustive search for the maximum PSNR over plus or minus the horizontal and vertical spatial uncertainties (in pixels) and plus or minus the temporal uncertainty (in frames). The processed video segment is fixed and the original video segment is shifted over the search range. For each spatial-temporal shift, a linear fit between the processed pixels and the original pixels is performed such that the mean squared error of (original - gain*processed + offset) is minimized (hence maximizing PSNR). Thus, this calculation of PSNR should yield PSNR values that are greater than or equal to commonly used PSNR implementations if the exhaustive search covered enough spatial-temporal shifts. The spatial-temporal search range and the amount of image cropping were performed in accordance with the calibration requirements given in the MM Experiment [1].

MATLAB code to compute PSNR is given in Section B.1 of Appendix B.

APPENDIX B: MATLAB CODE

This appendix contains MATLAB® code for many of the computations identified in this report. The following presents an ordered listing of the application of the MATLAB software routines to process the VQEG MM Phase I data.

1. Compute PSNR (see Section B.1).
2. Combine individual subjective data sets into one super-set (see Section B.3).
3. Fit each model's VM to the superset (see Section B.4).
4. Compute resolving power (see Section B.5).
5. Separate data if desired (e.g., discard clips with transmission errors).
6. Compute HRC averages if desired (see Section B.6).
7. Compute Pearson correlation, RMSE, and outlier ratio (see Section B.7).
8. Compute confidence intervals, if desired (see Section B.8).
9. Compute significant differences between models (see Section B.9).

The function "psnr_search.m" (Section B.1) calculates the PSNR for all processed video sequences (PVSs) from one VQEG MM Phase I experiment. See the function documentation for usage instructions.

The function "read_bigyuv.m" (Section B.2) reads frames from a raw big-YUV file.[8] Function "read_avi.m" (not provided) reads frames from an AVI file. These functions are used by psnr_search.m to read video clips into memory.

The script "common_map.m" (Section B.3) combines the individual subjective data sets into a single super-set on the MM ACR [5, 1] subjective scale, using the common set of video clips. This function needs all subjective data to be in one Microsoft Excel® file (i.e., one row for each clip). The columns are as follows: test, scene, HRC, meanDMOS, stDevDMOS, and NumVotes. All values (except for the header) need to be numbers, with no alphabetic characters. The common set needs to be at the end of each test. The tests must be listed from #1 at the top, then #2, etc. This correct row ordering can be produced by simply appending each test's Excel files one after the other.

The function "vqm_accuracy.m" (Section B.5) calculates *resolving power* at 95%, 90%, 75%, and 68%. This function takes one objective model's data at one resolution, and the subjective data superset calculated by function common_map.m. See the function interface for input and output variable definitions.

[8] A raw big-YUV file stores 4:2:2 YC_BC_R sampled video pixels (of one byte each) stored one row after another, where the first row of pixels are ordered as C_{B1}, Y_1, C_{R1}, Y_2, C_{B3}, Y_3, C_{R3}, Y_4, etc.

The other sections in this appendix contain code snippets that begin by defining variables of interest.

B.1 How to Calculate PSNR

```
function [results] = psnr_search(clip_dir, test, varargin)
% PSNR_SEARCH
%   Estimate the Y PSNR (PSNR) of all clips and HRCs in a video test where
%   the video clips are stored in the specified directory.  The video clips
%   must have names that conform to the standard naming convention
%   (test_scene_hrc.avi or test_scene_hrc.yuv, with no extra '_' or '.' in
%   the file names).  If AVI files are used, they must be in the 'YCbCr'
%   format.  A peak signal of 255 is used for calculation of PSNR.  This
%   routine can optionally utilize the means of a user-specified S-T block
%   size for the gain/offset calculation (rather than using the pixels to
%   perform this estimate).  Trying to fit too many points is very time
%   consuming for the MATLAB polyfit routine.  This routine uses double
%   precision calculations everywhere.
% SYNTAX
%   [results] = psnr_search('clip_dir', 'test', option);
% DESCRIPTION
%   This function will process all video clips in the user specified
%   clip_dir and test, estimate the Y PSNR of each clip, and then
%   average these clip results to produce an estimate for each HRC.  The
%   algorithm performs an exhaustive search for max PSNR over plus or minus
%   the spatial_uncertainty (in pixels) and plus or minus the
%   temporal_uncertainty (in frames).  The processed video segment is fixed
%   and the original video segment is shifted over the search range.  For
%   each spatial-temporal shift, a linear fit between the processed pixels
%   and the original pixels is performed such that the mean square error of
%   [original-gain*processed+offset] is minimized (hence maximizing PSNR).
%   For this version, the linear fit to obtain the gain and offset can be
%   optionally computed using the means of a user-specified S-T block size,
%   where the dimensions of the S-T block must be an integer divisor of the
%   sroi (spatial region of interest) and fstop-fstart+1 (temporal region
%   of interest).
%
%   Any or all of the following optional properties may be requested (the
%   first option is required for yuv files, but not for avi files since
%   this information is read from the avi header).
%
%   'yuv',rows,cols  Specifies the number of rows, cols, and fps for
%            yuv files.  The default is avi files.
%
%   'sroi',top,left,bottom,right,   Only use the specified spatial region
%                   of interest (sroi) for the PSNR
%                   calculation.  This is the sroi of the
%                   processed sequence, which remains fixed
%                   over all spatial shifts.  By default,
%                   sroi is the entire image reduced by the
%                   spatial uncertainty.
%
%   'frames',fstart,fstop  Only use the frames from fstart to fstop
%              (inclusive) to perform the PSNR estimate.  This
%              specifies the temporal segment of the processed
%              sequence, which remains fixed over all temporal
%              shifts.  By default, the temporal segment is the
%              entire file reduced by the temporal uncertainty.
%
%   'block',dx,dy,dt     The S-T block size, the mean of which is used to
%              estimate the gain and level offset.  By default,
%              dx=1, dy=1, dt=1 (one pixel) is used.  This
%              block size must be an integer divisor of sroi
%              and (fstop-fstart+1).  For large scenes, using
%              blocks instead of pixels (e.g., dx=4, dy=4,
%              dt=1) can greatly increase the run speed and
%              reduce the memory requirements with very little
%              reduction in accuracy.  PSNR is still calculated
%              using the pixels when this option is selected.
%
%   'spatial_uncertainty',x,y   Specifies the spatial uncertainty (plus
%                 or minus, in pixels) over which to
%                 search.  The processed remains fixed and
%                 the original is shifted.  By default,
%                 this is set to zero.
%
%   'temporal_uncertainty',t  Specifies the temporal uncertainty
%               (plus or minus, in frames) over which
%               to search.  The processed remains fixed
%               and the original is shifted.  By
%               default, this is set to zero.
%
%   'verbose'   Display output during processing.
%
%
%   The returned variable [results] is a struct that contains the following
%   information for each processed clip i:
%
%   results(i).test  The test name for the video clip.
%   results(i).scene   The scene name for the video clip.
%   results(i).hrc   The HRC name for the video clip.
%   results(i).yshift  The y shift for max PSNR.
%   results(i).xshift  The x shift for max PSNR.
%   results(i).tshift  The time shift for max PSNR.
%   results(i).gain  The gain*processed+offset for max PSNR.
%   results(i).offset
%   results(i).psnr  The maximum PSNR observed over the search.
%
% EXAMPLES
%   These examples illustrate how to call the routine to process the VQEG
%   MM Phase I test scenes, where test scenes from each subjective
%   experiment are stored in a unique directory.
%
%   q01 = psnr_search('d:\avi_q01\','q01','sroi',5,5,140,172,...
%     'spatial_uncertainty',1,1,'temporal_uncertainty',8,'verbose');
%   c01 = psnr_search('d:\avi_c01\','c01','sroi',8,8,281,345,...
%     'spatial_uncertainty',1,1,'temporal_uncertainty',8,'verbose');
%   v01 = psnr_search('e:\avi_v01\','v01','sroi',14,14,467,627,'block',...
%     2,2,1,'spatial_uncertainty',1,1,'temporal_uncertainty',8,'verbose');
%

% Define the peak signal level
peak = 255.0;

% Add extra \ in clip_dir in case user did not
clip_dir = strcat(clip_dir,'\');

% Validate input arguments and set their defaults
is_yuv = 0;
is_whole_image = 1;
is_whole_time = 1;
x_uncert = 0;
y_uncert = 0;
t_uncert = 0;
verbose = 0;
dx=1;
dy=1;
dt=1;
cnt=1;
while cnt <= length(varargin),
  if ~isstr(varargin{cnt}),
    error('Property value passed into psnr_search is not recognized');
  end
  if strcmpi(varargin(cnt),'yuv') == 1
    rows = varargin{cnt+1};
    cols = varargin{cnt+2};
    is_yuv = 1;
    cnt = cnt + 3;
  elseif strcmpi(varargin(cnt),'sroi') == 1
    top = varargin{cnt+1};
    left = varargin{cnt+2};
    bottom = varargin{cnt+3};
    right = varargin{cnt+4};
    is_whole_image = 0;
    cnt = cnt + 5;
  elseif strcmpi(varargin(cnt),'frames') == 1
    fstart = varargin{cnt+1};
    fstop = varargin{cnt+2};
    is_whole_time = 0;
    cnt = cnt + 3;
  elseif strcmpi(varargin(cnt),'block')== 1
    dx = varargin{cnt+1};
    dy = varargin{cnt+2};
    dt = varargin{cnt+3};
    cnt = cnt +4;
  elseif strcmpi(varargin(cnt), 'spatial_uncertainty') ==1
    x_uncert = varargin{cnt+1};
    y_uncert = varargin{cnt+2};
    cnt = cnt + 3;
  elseif strcmpi(varargin(cnt), 'temporal_uncertainty') ==1
    t_uncert = varargin{cnt+1};
    cnt = cnt + 2;
  elseif strcmpi(varargin(cnt),'verbose') == 1
    verbose = 1;
    cnt = cnt +1;
  else
    error('Property value passed into psnr_search not recognized');
  end
end

%  Get a directory listing
files = dir(clip_dir);  % first two files are '.' and '..'
num_files = size(files,1);

% Find the HRCs and their scenes for the specified video test
hrc_list = {};
scene_list = {};
for i=3:num_files
  this_file = files(i).name;
  und = strfind(this_file,'_'); % find underscores and period
  dot = strfind(this_file,'.');
  if(size(und,2)==2) % possible standard naming convention file found
    this_test = this_file(1:und(1)-1);  % pick off the test name
    if(strmatch(test,this_test,'exact'))  % test clip found
      this_scene = this_file(und(1)+1:und(2)-1);
      this_hrc = this_file(und(2)+1:dot(1)-1);
      % See if this HRC already exists and find its list location
      loc = strmatch(this_hrc,hrc_list,'exact');
      if(loc)  % HRC already present, add to scene list for that HRC
        if(size(strmatch(this_scene,scene_list{loc},'exact'),1)==0)
          scene_list{loc} = [scene_list{loc} this_scene];
        end
      else  % new HRC found
        hrc_list = [hrc_list;{this_hrc}];
        this_loc = size(hrc_list,1);
        scene_list(this_loc) = {{this_scene}};
      end
    end
  end
end

scene_list = scene_list';
num_hrcs = size(hrc_list,1);

%Results struct to store results, shifts are how much the original must be
%shifted with respect to the processed
results = struct('test', {}, 'scene', {}, 'hrc', {}, 'yshift', {}, ...
  'xshift', {}, 'tshift', {}, 'gain', {}, 'offset', {}, 'psnr', {});
```

```
% Process one HRC at a time to compute average PSNR for that HRC
index = 1;  % index to store results
for i = 1:num_hrcs

 psnr_ave = 0;  % initialize the psnr average summer for this HRC
 this_hrc = hrc_list{i};
 if(strmatch('original',this_hrc,'exact')) % Don't process original
  continue;
 end
 num_scenes = size(scene_list{i},2);  % Number of scenes in this HRC

 for j = 1:num_scenes

  this_scene = scene_list{i}{j};
  results(index).test = test;
  results(index).scene = this_scene;
  results(index).hrc = this_hrc;

  % Read original and processed video files
  if (~is_yuv)  % AVI file
   % Re-generate the original and processed avi file names
   orig = strcat(clip_dir, test,'_', this_scene, '_', 'original', '.avi');
   proc = strcat(clip_dir, test,'_', this_scene, '_', this_hrc, '.avi');
   [avi_info] = read_avi('Info',orig);
   rows = avi_info.Height;
   cols = avi_info.Width;
   % Set/Validate the ROI
   if (is_whole_image) % make ROI whole image less uncertainty
    top = 1+y_uncert;
    left = 1+x_uncert;
    bottom = rows-y_uncert;
    right = cols-x_uncert;
   elseif (top<1 || left<1 || bottom>rows || right>cols)
    display('Requested ROI is too large for image size.\n');
    return
   end
   tframes = avi_info.NumFrames;  % total frames in file
   % Set/Validate the time segment to use
   if (is_whole_time) % use whole time segment less uncertainty
    fstart= 1+t_uncert;
    fstop = tframes-t_uncert;
   elseif (fstart<1 || fstop>tframes)
    display('Temporal segment too large for file size.\n');
    return
   end
   % Validate the spatial uncertainty search bounds
   if (left-x_uncert < 1 || right+x_uncert > cols)
    display('Spatial x-uncertainty too large.\n');
    return;
   end
   if (top-y_uncert < 1 || bottom+y_uncert > rows)
    display('Spatial y-uncertainty too large.\n');
    return;
   end
   % Validate the temporal uncertainty search bounds
   if(fstart-t_uncert < 1 || fstop+t_uncert > tframes)
    display('Temporal uncertainty too large.\n');
    return;
   end
   % Validate the block size in the x-direction
   if (mod(right-left+1,dx) ~= 0)
    fprintf('Block x-size must divide %i\n',right-left+1);
    return
   end
   % Validate the block size in the y-direction
   if (mod(bottom-top+1,dy) ~= 0)
    fprintf('Block y-size must divide %i\n',bottom-top+1);
    return
   end
   % Validate the block size in the t-direction
```

```
    if (mod(fstop-fstart+1,dt) ~= 0)
      fprintf('Block t-size must divide %i\n',fstop-fstart+1);
      return
    end
    % Read in video and clear color planes to free up memory
    [y_orig,cb,cr] = read_avi('YCbCr',orig,'frames',fstart-t_uncert,...
      fstop+t_uncert, 'sroi',top-y_uncert,left-x_uncert,...
      bottom+y_uncert,right+x_uncert);
    clear cb cr;
    [y_proc,cb,cr] = read_avi('YCbCr',proc,'frames',fstart,fstop,...
      'sroi',top,left,bottom,right);
    clear cb cr;
  else  % YUV file
    % Re-generate the original and processed YUV file name
    orig = strcat(clip_dir, test,'_', this_scene, '_', 'original', '.yuv');
    proc = strcat(clip_dir, test,'_', this_scene, '_', this_hrc, '.yuv');
    % Set/Validate the ROI
    if (is_whole_image) % make ROI whole image less uncertainty
      top = 1+y_uncert;
      left = 1+x_uncert;
      bottom = rows-y_uncert;
      right = cols-x_uncert;
    elseif (top<1 || left<1 || bottom>rows || right>cols)
      display('Requested ROI too large for image size.\n');
      return;
    end
    % Find the total frames of the input file
    [fid, message] = fopen(orig, 'r');
    if fid == -1
      fprintf(message);
      error('Cannot open this clip''s bigyuv file, %s', orig);
    end
    % Find last frame.
    fseek(fid,0, 'eof');
    tframes = ftell(fid) / (2 * rows * cols);
    fclose(fid);
    % Set/Validate the time segment to use
    if (is_whole_time) % use whole time segment less uncertainty
      fstart= 1+t_uncert;
      fstop = tframes-t_uncert;
    elseif (fstart<1 || fstop>tframes)
      display('Temporal segment too large for file size.\n');
      return
    end
     % Validate the spatial uncertainty search bounds
    if (left-x_uncert < 1 || right+x_uncert > cols)
      display('Spatial x-uncertainty too large.\n');
      return;
    end
    if (top-y_uncert < 1 || bottom+y_uncert > rows)
      display('Spatial y-uncertainty too large.\n');
      return;
    end
    % Validate the temporal uncertainty search bounds
    if(fstart-t_uncert < 1 || fstop+t_uncert > tframes)
      display('Temporal uncertainty too large.\n');
      return;
    end
    % Validate the block size in the x-direction
    if (mod(right-left+1,dx) ~= 0)
      fprintf('Block x-size must divide %i\n',right-left+1);
      return
    end
    % Validate the block size in the y-direction
    if (mod(bottom-top+1,dy) ~= 0)
      fprintf('Block y-size must divide %i\n',bottom-top+1);
      return
    end
    % Validate the block size in the t-direction
    if (mod(fstop-fstart+1,dt) ~= 0)
      fprintf('Block t-size must divide %i\n',fstop-fstart+1);
      return
    end
    % Read in video and clear color planes to free up memory
    [y_orig,cb,cr] = read_bigyuv(orig,'frames',fstart-t_uncert,...
      fstop+t_uncert,'size',rows,cols,'sroi',top-y_uncert,...
      left-x_uncert,bottom+y_uncert,right+x_uncert);
    clear cb cr;
    [y_proc,cb,cr] = read_bigyuv(proc,'frames',fstart,fstop,...
      'size',rows,cols,'sroi',top,left,bottom,right);
    clear cb cr;
  end

  % Convert images to double precision
  y_orig = double(y_orig);
  y_proc = double(y_proc);

  [nrows, ncols, nsamps] = size(y_proc);
  % Reshape y_proc for the PSNR calculation:  this stays fixed
  y_proc = reshape(y_proc,nrows*ncols*nsamps,1); % make column vector

  %  Setup temp proc array for gain calculation
  if (dy~=1 || dx~=1 || dt~=1)
    % Compute mean over S-T blocks and organize as column vector
    % Sum over dy
    y_proc_mean = sum(reshape(y_proc,dy,nrows*ncols*nsamps/dy),1);
    y_proc_mean = permute(reshape(y_proc_mean,nrows/dy,ncols,nsamps),[2 3 1]);
    % Sum over dx
    y_proc_mean = ...
      sum(reshape(y_proc_mean,dx,nrows*ncols*nsamps/(dy*dx)),1);
    y_proc_mean = permute(reshape(y_proc_mean,ncols/dx,nsamps,nrows/dy),[2 3 1]);
    % Sum over dt
    y_proc_mean = ...
      sum(reshape(y_proc_mean,dt,nrows*ncols*nsamps/(dy*dx*dt)),1);
    y_proc_mean = ...
      permute(reshape(y_proc_mean,nsamps/dt,nrows/dy,ncols/dx),[2 3 1]);
    y_proc_mean = ...
      reshape(y_proc_mean,nrows*ncols*nsamps/(dy*dx*dt),1)/(dy*dx*dt);
  end

  % Compute PSNR for each spatial-temporal shift
  best_psnr = -inf;
  best_xshift = 0;
  best_yshift = 0;
  best_tshift = 0;
  best_gain = 1;
  best_offset = 0;
  if(verbose)
    fprintf('\nTest = %s,   Scene = %s,   HRC = %s\n',...
      test, this_scene, this_hrc);
  end

  for k = -t_uncert:t_uncert
    for m = -x_uncert:x_uncert
      for n = -y_uncert:y_uncert

        % Perform gain and level offset calculation
        if (dy==1 && dx==1 && dt==1) % use the pixels
          this_fit = ...
            polyfit(y_proc,reshape(y_orig(1+n+y_uncert:n+y_uncert+nrows,...
            1+m+x_uncert:m+x_uncert+ncols,...
            1+k+t_uncert:k+t_uncert+nsamps),...
            nrows*ncols*nsamps,1),1);
        else % use the means of S-T blocks
          % Sum over dy
          y_orig_mean = ...
            sum(reshape(y_orig(1+n+y_uncert:n+y_uncert+nrows,...
            1+m+x_uncert:m+x_uncert+ncols,...
            1+k+t_uncert:k+t_uncert+nsamps),dy,...
            nrows*ncols*nsamps/dy),1);
          y_orig_mean = ...
            permute(reshape(y_orig_mean,nrows/dy,ncols,nsamps),[2 3 1]);
```

```
            % Sum over dx
            y_orig_mean = ...
              sum(reshape(y_orig_mean,dx,nrows*ncols*nsamps/(dy*dx)),1);
            y_orig_mean = ...
              permute(reshape(y_orig_mean,ncols/dx,nsamps,nrows/dy),[2 3 1]);
            % Sum over dt
            y_orig_mean = ...
                 sum(reshape(y_orig_mean,dt,nrows*ncols*nsamps/(dy*dx*dt)),1);
            y_orig_mean = permute(reshape(y_orig_mean, ...
                 nsamps/dt,nrows/dy,ncols/dx),[2 3 1]);
            y_orig_mean = reshape(y_orig_mean, ...
                 nrows*ncols*nsamps/(dy*dx*dt),1)/(dy*dx*dt);
            this_fit = polyfit(y_proc_mean,y_orig_mean,1);
          end

          % Calculate the PSNR
          this_psnr = 10*(log10(peak*peak)- ...
               log10(sum(((this_fit(1)*y_proc+this_fit(2))-...
            reshape(y_orig(1+n+y_uncert:n+y_uncert+nrows, ...
               1+m+x_uncert:m+x_uncert+ncols,...
            1+k+t_uncert:k+t_uncert+nsamps), ...
               nrows*ncols*nsamps,1)).^2)/(nrows*ncols*nsamps)));
          if(this_psnr > best_psnr)
            best_psnr = this_psnr;
            best_yshift = n;
            best_xshift = m;
            best_tshift = k;
            best_gain = this_fit(1);
            best_offset = this_fit(2);
            if(verbose)
              fprintf('dy =%3i, dx =%3i, dt =%3i, gain = %5.4f, offset = %5.4f, PSNR = %5.4f\n',...
                best_yshift,best_xshift,best_tshift,best_gain,best_offset,best_psnr);
            end
          end

        end
      end
    end

    results(index).yshift = best_yshift;
    results(index).xshift = best_xshift;
    results(index).tshift = best_tshift;
    results(index).gain = best_gain;
    results(index).offset = best_offset;
    results(index).psnr = best_psnr;
    psnr_ave = psnr_ave+best_psnr;
    index = index+1;

  end

  % Compute average PSNR for this HRC
  psnr_ave = psnr_ave/(num_scenes);
  if(verbose)
    fprintf('HRC = %s, psnr_ave = %5.4f\n',this_hrc, psnr_ave);
  end

end
```

B.2 How To Read Big-YUV Files

```
function [y,cb,cr] = read_bigyuv(file_name, varargin);
% READ_BIGYUV
%   Read images from bigyuv-file.
% SYNTAX
%   [y] = read_bigyuv(file_name);
%   [y,cb,cr] = read_bigyuv(...);
%   [...] = read_bigyuv(...,'PropertyName',PropertyValue,...);
% DESCRIPTION
%   Read in images from bigyuv file named 'file_name'.
```

```
%
%   The luminance plane is returned in 'Y'; the color planes are
%   returned in 'cb' and 'cr' upon request.  The Cb and Cr color planes
%   will be upsampled by 2 horizontally.
%
%   The following optional properties may be requested:
%
%   'sroi',top,left,bottom,right,
%                       Spatial region of interest to be returned.  By default,
%                       the entirety of each image is returned.
%                       Inclusive coordinates (top,left),(bottom,right) start
%                       numbering with row/line number 1.
%   'size',row,col,    Size of images (row,col).  By default, row=486,
%                           col=720.
%   'frames',start,stop,    Specify the first and last frames, inclusive,
%                                   to be read ('start' and 'stop').  By
%                                   default, the first frame is read.
%   '128'        Subtract 128 from all Cb and Cr values.  By default, Cb
%                       and Cr values are left in the [0..255] range.
%   'interp'           Interpolate Cb and Cr values.  By default, color
%                       planes are pixel replicated.  Note:  Interpolation is slow.
%
%   Color image pixels will be pixel replicated, so that Cb and Cr images
%   are not subsampled by 2 horizontally.

% read values from clip_struct that can be over written by variable argument
% list.
is_whole_image = 1;
is_sub128 = 0;
is_interp = 0;

num_rows = 486;
num_cols = 720;

start = 1;
stop = 1;

% parse varargin list (property values)
cnt = 1;
while cnt <= nargin - 1,
    if ~isstr(varargin{cnt}),
        error('Property value passed into bigyuv_read not recognized');
    end
    if strcmp(lower(varargin(cnt)),'sroi') == 1,
        sroi.top = varargin{cnt+1};
        sroi.left = varargin{cnt+2};
        sroi.bottom = varargin{cnt+3};
        sroi.right = varargin{cnt+4};
        is_whole_image = 0;
        cnt = cnt + 5;
    elseif strcmp(lower(varargin(cnt)),'size') == 1,
        num_rows = varargin{cnt+1};
        num_cols = varargin{cnt+2};
        cnt = cnt + 3;
    elseif strcmp(lower(varargin(cnt)),'frames') == 1,
        start = varargin{cnt+1};
        stop = varargin{cnt+2};
        cnt = cnt + 3;
    elseif strcmp(lower(varargin(cnt)),'128') == 1,
        is_sub128 = 1;
        cnt = cnt + 1;
    elseif strcmp(lower(varargin(cnt)),'interp') == 1,
        is_interp = 1;
        cnt = cnt + 1;
    else
        error('Property value passed into bigyuv_read not recognized');
    end
end

if mod(num_cols,2) ~= 0,
    fprintf('Error: number of columns must be an even number.\nThis 4:2:2 format sores 4 bytes
for each 2 pixels\n');
    error('Invalid specification for argument "num_cols" in read_bigyuv');
end

% Open image file
% [test_struct.path{1} clip_struct.file_name{1}]
[fid, message] = fopen(file_name, 'r');
if fid == -1
    fprintf(message);
    error('bigyuv_read cannot open this clip''s bigyuv file, %s', file_name);
end

% Find last frame.
fseek(fid,0, 'eof');
total = ftell(fid) / (2 * num_rows * num_cols);
if stop > total,
    error('Requested a frame past the end of the file.  Only %d frames available', total);
end
if stop < 0,
    error('Range of frames invalid');
end
if start > stop | stop < 1,
    error('Range of frames invalid, or no images exist in this bigyuv file');
end

% find range of frames requested.
prev_tslice_frames = start - 1;
tslice_frames = stop - start + 1;
number = start;

% go to requested location
if isnan(start),
    error('first frame of this clip is undefined (NaN).');
end
offset = prev_tslice_frames * num_rows * num_cols * 2; %pixels each image
status = fseek(fid, offset, 'bof');

if status == -1,
    fclose(fid);
    error('bigyuv_read cannot seek requested image location');
end

% initialize memory to hold return images.
y = zeros(num_rows,num_cols,tslice_frames, 'single');

if (nargout == 3),
    cb = y;
    cr = y;
end

% loop through & read in the time-slice of images
this_try = 1;
for cnt = 1:tslice_frames,
    where = ftell(fid);
    [hold_fread,count] = fread(fid, [2*num_cols,num_rows], 'uint8=>uint8');
    if count ~= 2*num_cols*num_rows,
        % try one more time.
        fprintf('Warning: bigyuv_read could not read entirety of requested image time-slice; re-
trying\n');
        %pause(5);
        if where == -1,
            fprintf('Could not determine current location.  Re-try failed.\n');
            error('bigyuv_read could not read entirety of requested image time-slice');
            fclose(fid);
        end
        fseek(fid, where, 'bof');
        [hold_fread,count] = fread(fid, [2*num_cols,num_rows], 'uint8=>uint8');
        if count ~= 2*num_cols*num_rows,
            fclose(fid);
            hold = sprintf('time-slice read failed for time-slice in %s\nbigyuv_read could not
read entirety of requested time-slice', file_name);
            error(hold);
        end
    end

    % pick off the Y plane (luminance)
    temp = reshape(hold_fread', num_rows, 2, num_cols);
    uncalib = squeeze(temp(:,2,:));
    y(:,:,cnt) = single(uncalib);

    % If color image planes are requested, pick those off and perform
    % pixel replication to upsample horizontally by 2.
    if nargout == 3,
        temp = reshape(hold_fread,4,num_rows*num_cols/2);

        color = reshape(temp(1,:),num_cols/2,num_rows)';
        color2 = [color ; color];
        uncalib = reshape(color2,num_rows,num_cols);
        cb(:,:,cnt) = single(uncalib);
        if is_sub128,
            cb(:,:,cnt) = cb(:,:,cnt) - 128;
        end

        color = reshape(temp(3,:),num_cols/2,num_rows)';
        color2 = [color ; color];
        uncalib = reshape(color2,num_rows,num_cols);
        cr(:,:,cnt) = single(uncalib);
        if is_sub128,
            cr(:,:,cnt) = cr(:,:,cnt) - 128;
        end

        % Interpolate, if requested
        if is_interp == 1,
            for i=2:2:num_cols-2,
                cb(:,i,cnt) = (cb(:,i-1,cnt) + cb(:,i+1,cnt))/2;
                cr(:,i,cnt) = (cr(:,i-1,cnt) + cr(:,i+1,cnt))/2;
            end
        end
    end
end

fclose(fid);

if ~is_whole_image,
    y = y(sroi.top:sroi.bottom, sroi.left:sroi.right, :);
    if nargout == 3,
        cb = cb(sroi.top:sroi.bottom, sroi.left:sroi.right, :);
        cr = cr(sroi.top:sroi.bottom, sroi.left:sroi.right, :);
    end
end
```

B.3 How To Map Individual Experiments to the Superset using Common Set Clips

```
%  Script common_map.m
%
%  Script to estimate the data mapping to combine subjective data sets
%  within one resolution (e.g., QCIF), for the VQEG MM data.  A linear fit
%  will determine the map, where the common scores for a given test are
%  used for the independent variable x and the average of the common scores
%  over all tests is used for the dependent variable y.
%
%  Import subjective data manually from XLS files (using the File -> Import
%  Data menus in the main MATLAB window).  When prompted, accept the default
%  variable names for the data (data).  The XLS files are sorted by test,
%  scene, and then HRC, except that the common sources are at the end of
%  each test.  This routine assumes that the XLS file has been modified
%  to only contain test, scene, hrc, meanMOS, stdDevMOS, and NumVotes in
%  that order, and that the test has been converted to a number (i.e., no
%  letters).  This allows Matlab to import the data properly into a data
%  matrix.
%
n = 166;  % total number of clips per test, including originals
nc = 30;  % number of common clips at end of each test

% =1 to include original clips in mapping, =0 to discard
% Should discard for DMOS but could keep for MOS
keep_originals = 0;

%  Determine the number of tests
ntests = size(data,1)/n;

%  Loop through tests, if the subjective scores for all common video clips
%  in the test are valid, then that test will be included in the data map.
%  The first column of data is assumed to have the test number.
test_num = [];
common = [];
for i=1:ntests
    this_test = data(1+n*(i-1),1);
    this_common = data(1+n*i-nc:n*i, :);
    if (~keep_originals)
        this_common = this_common(find(this_common(:,3) ~= 0),4); % pick off mos only
    else
        this_common = this_common(:, 4);
    end
    % Make sure that all mos scores are valid before using this set
    if (~isnan(this_common))  % There are no NaN for MOS in common set
        test_num = [test_num this_test];
        common = [common this_common];
    end
end

common_mean = mean(common,2);
valid_tests = size(test_num,2);

% Fit and plot
porder = 1;  % polynomial order
map_fits = [];  % holds the polynomial fit
map_corrs = [];  % holds the correlation coefficents between individual test and mean test
x = 1:0.1:5;
for i = 1:ntests
    this_fit = polyfit(common(:,i),common_mean,porder)';
    map_fits = [map_fits this_fit];
    y = polyval(this_fit,x);
    r = corrcoef(common(:,i),common_mean);
    map_corrs = [map_corrs r(1,2)];
    fprintf('Test Number = %i, correlation = %f\n',test_num(i), r(1,2))
    plot(common(:,i),common_mean, '.')
    grid on
    xlabel('Individual Test');
    ylabel('Mean Test');
    hold on
    plot(x,y,'r');
    hold off
    pause
end

% Apply the common map to the entire subjective data set
% For a linear fit, the stdev scales as the gain term of the fit.
scaled_mos = [];  % all arrays of dimension n x valid_tests
unscaled_mos = [];
scaled_std = [];
unscaled_std = [];
num_viewers = [];
for i=1:ntests
    exclude_orig = find(data(:,1)==test_num(i) & data(:,3)~=0);
    include_orig = find(data(:,1)==test_num(i));
    % This std scaling is only valid for linear fits, approximate otherwise
    if (~keep_originals)
        scaled_mos(:,i) = polyval(map_fits(:,i),data(exclude_orig, 4));
        unscaled_mos(:,i) = data(exclude_orig, 4);
        scaled_std(:,i) = sum(map_fits(1:porder,i))*data(exclude_orig, 5);
        unscaled_std(:,i) = data(exclude_orig, 5);
        num_viewers(:,i) = data(exclude_orig, 6);
    else
        scaled_mos(:,i) = polyval(map_fits(:,i),data(include_orig, 4));
        unscaled_mos(:,i) = data(include_orig, 4);
        scaled_std(:,i) = sum(map_fits(1:porder,i))*data(include_orig, 5);
        unscaled_std(:,i) = data(include_orig, 5);
        num_viewers(:,i) = data(include_orig, 6);
    end
end
```

B.4 How To Fit Each Model to the Superset

```
% Given the following variables:
% 'vqm' is a column vector that holds one objective model scores for the superset.
% 'vqm_sign' = 1 or -1 and gives the direction of 'vqm' with respect to the subjective scale.
%    Higher values of vqm for higher subjective scores means that 'vqm_sign' = 1.
% 'mos' is a column vector that holds the corresponding MOS values for the superset.
% 'order' specifies the order of the monotonic polynomial fit (e.g., order=3).
%
% Following code implements monotonic polynomial fitting using the MATLAB Optimization
% Toolbox routine lsqlin.
%

num_comb = length(vqm);  % total number of clips in the superset

% Create x and dx arrays.  The dx slope array (holds the derivatives of mos with respect
% to vqm), the vqm_sign specifies the direction of the slope that must not change over the
% vqm range.
x = ones(num_comb,1);
dx = zeros(num_comb,1);
for col = 1:order
    x = [x vqm.^col];
    dx = [dx col*vqm.^(col-1)];
end

% The lsqlin routine uses <= inequalities.  Thus, if vqm_sign is -1 (negative slope),
% we are correct but if vqm_sign is +1 (positive slope), we must multiply each side by -1.
if (vqm_sign == 1)
    dx = -1*dx;
end

% Perform the fit and organize to what would have been output by polyfit
fit = lsqlin(x,mos,dx,zeros(num_comb,1));
fit = flipud(fit)';

% Use polyval to find the fitted vqm values (e.g., vqm_hat)
vqm_hat = polyval(fit,vqm);
```

B.5 How To Compute Resolving Power

```
function [resolving_power] = vqm_accuracy (vqm, num_viewers, mos, std, deg_of_freedom)
% MATLAB function [resolving_power] = ...
%       vqm_accuracy (vqm, num_viewers, mos, std, deg_of_freedom)
%
% Compute resolving power for one model.
%
%    vqm is the video quality metric score for this src_id x hrc_id
%    num_viewers is the number of viewers that rated this src_id x hrc_id
%    mos is the mean opinion score of this src_id x hrc_id
%    std is the standard-deviation of this src_id x hrc_id
%
```

```
% All of the above arrays must be the same length.  The VQM must already be
% fitted to the MOS.
%
%   deg_of_freedom is the number of degrees of freedom for the fit between
%           VQM and MOS prior to calling this routine.
%
% returned data contains:
%   resolving_power(1) = 95% Resolving Power
%   resolving_power(2) = 90% Resolving Power
%   resolving_power(3) = 75% Resolving Power
%   resolving_power(4) = 68% Resolving Power

variance = std.^2;
num_comb = length(vqm);

% Perform the vqm RMSE calculation using vqm.
vqm_rmse = (sum((vqm-mos).^2)/(num_comb - deg_of_freedom))^0.5;

% Perform the vqm resolution measurement using both vqm and mos.
vqm_pairs = repmat(vqm,1,num_comb)-repmat(vqm',num_comb,1);
mos_pairs = repmat(mos,1,num_comb)-repmat(mos',num_comb,1);
stand_err_diff = sqrt(repmat(variance./num_viewers,1,num_comb)+ ...
    repmat((variance./num_viewers)',num_comb,1));
z_pairs = mos_pairs./stand_err_diff;

% Include everything above the diagonal.
delta_vqm = [];
z = [];
for col = 2:num_comb
    delta_vqm = [delta_vqm; vqm_pairs(1:col-1,col)];
    z = [z; z_pairs(1:col-1,col)];
end

% Switch on z and delta_vqm for negative delta_vqm
z_vqm = z;
negs_vqm = find(delta_vqm < 0);
delta_vqm(negs_vqm) = -delta_vqm(negs_vqm);
z_vqm(negs_vqm) = -z_vqm(negs_vqm);


% Compute the average confidence that vqm(2) is worse than vqm(1) in mean_cdf_z_vqm.
cdf_z_vqm = .5+erf(z_vqm/sqrt(2))/2;

% One control parameter for delta_vqm resolution plot; number of vqm bins,
% equally spaced from min(delta_vqm) to max(delta_vqm).

% Sliding neighborhood filter with 50% overlap means that there will actually
% be vqm_bins*2-1 points on the delta_vqm resolution plot.
vqm_bins = 10; % How many bins to divide full vqm range for local averaging
vqm_low = min(delta_vqm); % lower limit on delta_vqm
vqm_high = max(delta_vqm); % upper limit on delta_vqm
vqm_step = (vqm_high-vqm_low)/vqm_bins; % size of delta_vqm bins

% lower, upper, and center bin locations
low_limits = [vqm_low:vqm_step/2:vqm_high-vqm_step];
high_limits = [vqm_low+vqm_step:vqm_step/2:vqm_high];
centers = [vqm_low+vqm_step/2:vqm_step/2:vqm_high-vqm_step/2];

mean_cdf_z_vqm = zeros(1,2*vqm_bins-1);
for i=1:2*vqm_bins-1
    in_bin = find(low_limits(i) <= delta_vqm & delta_vqm < high_limits(i));
    mean_cdf_z_vqm(i) = mean(cdf_z_vqm(in_bin));
end

% % Optional code to plot resolving power curve.
% % The x-axis is vqm(2)-vqm(1).  The Y-axis is always the average
% % confidence that vqm(2) is worse than vqm(1).
% figure(1)
% plot(centers,mean_cdf_z_vqm)
% grid
% set(gca,'LineWidth',1)
% set(gca,'FontName','Ariel')
% set(gca,'fontsize',11)
% xlabel('VQM (2) - VQM (1)')
% ylabel('Average Confidence VQM (2) is worse than VQM (1)')
% title('VQM Resolving Power')

% Compute each resolving power by interpolating the mean_cdf_z_vqm graph

% 95% resolving power
i = length(centers) - 1;
while mean_cdf_z_vqm(i) > 0.95 && i > 1,
    i = i -1;
end
j = min(length(centers), i+1);
resolving_power(1) = interp1(mean_cdf_z_vqm(i:j),centers(i:j), 0.95);

% 90% resolving power
i = length(centers) - 1;
while mean_cdf_z_vqm(i) > 0.90 && i > 1,
    i = i -1;
end
j = min(length(centers), i+1);
resolving_power(2) = interp1(mean_cdf_z_vqm(i:j),centers(i:j), 0.90);

% 75% resolving power
i = length(centers) - 1;
while mean_cdf_z_vqm(i) > 0.75 && i > 1,
    i = i -1;
end
j = min(length(centers), i+1);
resolving_power(3) = interp1(mean_cdf_z_vqm(i:j),centers(i:j), 0.75);

% 68% resolving power
i = length(centers) - 1;
while mean_cdf_z_vqm(i) > 0.68 && i > 1,
    i = i -1;
end
j = min(length(centers), i+1);
resolving_power(4) = interp1(mean_cdf_z_vqm(i:j),centers(i:j), 0.68);

% return infinity if can't compute
resolving_power(isnan(resolving_power)) = inf;
```

B.6 How to Compute HRC Averages

```
%  Given the following variables:
%  ‘all_hrcs’ is a cell array with each HRC name (each occurring once, only)
%  ‘all_is_hrc’ is a cell array with one entry for each clip, containing that clip’s HRC
%  ‘all_super_mos’ is a cell array with one entry for each clip, containing that
%   clip’s super-set DMOS score.
%  ‘all_super_std’ is an array with one entry for each clip, containing that
%   clip’s super-set standard deviation of MOS scores.
%  ‘all_super_viewers’ is an array with one entry for each clip, containing that
%   clip’s super-set number of viewers.
%  ‘all_super_obj’ is a 2-D array with one entry for each objective model (2nd dimension) and
%   one entry for each clip (1st dimension), containing that model’s VM score for the
%   current clip (after fitting).
%
%   Compute the following variables:

     j = 1;
    for i=1:length(all_hrcs),
        if length(all_hrcs{i}) > 3,
            curr = find(strcmp(all_is_hrc,all_hrcs{i}));
            if length(curr) ~= 8,
                error('Didn''t find all 8 SRC for this HRC');
            end
            super_mos(j) = mean(all_super_mos(curr));
            for k = 1:(size(all_super_obj,2)),
                super_obj(j,k) = mean(all_super_obj(curr,k));
            end
            super_viewers(j) = sum(all_super_viewers(curr));
            super_std(j) = sqrt( mean( (all_super_std(curr)).^2  ));

            j = j + 1;
        end
    end
```

B.7 How to Compute Pearson Correlation, RMSE, and Outlier Ratio

```
% Given the following variables:
% ‘super_mos’ holds the super-set’s DMOS scores.
% ‘super_std’ holds the super-set’s standard deviation of DMOS scores.
% ‘super_viewers’ holds the number of viewers used to compute each data
%  point in super_mos and super_std.
% ‘yhat’ holds one model VM fitted to this super-set
% ‘len_minus_df’ is the number of data points minus the degrees of freedom
%  of the fit, adjusted by any averaging performed (e.g., HRC averaging).
%  If computed on a per-clip basis, then this is simply the
%  number of clips (e.g., 152 for one VQEG MM experiment) minus 4 (three degrees
%  of freedom for the fit variables, plus one for the constant term).

% compute correlation, and place that into variable ‘corr’
temp = corrcoef(yhat, super_mos);
corr = temp(1,2);

% compute RMSE, and place that into variable ‘rmse’
rmse = sqrt(sum((yhat - super_mos).^2) / len_minus_df );

% compute Outlier Ratio, and place that into variable ‘outratio’
temp = 2.069 * super_std ./ sqrt(super_veiwers);
outratio = length(find(abs(yhat - super_hat) > temp)) / length(yhat);

%  When computing on a per-HRC basis, the above equations remain the same but
%  the definitions of the variables change slightly.  ‘len_minus_df’ must be
%  divided by the number of SRC averaged (e.g., 8 for VQEG MM).
%  Also, the number of viewers increases by the number of SRC averaged.  This
%  also changes the 2.069 multiplier constant to 1.96 in the equation for temp.
```

B.8 How to Compute Confidence Intervals

```
function [corr_ci_lower, corr_ci_upper, rmse_lower, rmse_upper, ...
       or_lower, or_upper] = get_confidence(corrs, num_clips, rmses, outratios)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compute the MM test plan / MM final report's confidence interval on
% statistics correlation, rmse & outlier ratio; on a per-clip basis.
% ‘corrs’ is an array holding each model’s correlation
% ‘num_clips’ is the number of clips used in the calculation
% ‘rmses’ is an array holding each model’s RMSE
% ‘outratios’ is an array holding each model’s outlier ratio
%
% The lower and upper confidence bounds are returned.

temp = 0.5 * log( (1 + corrs) ./ (1 - corrs) );
corr_ci_lower = temp - 1.96 * sqrt(1 / (num_clips - 3));
corr_ci_upper = temp + 1.96 * sqrt(1 / (num_clips - 3));
corr_ci_lower = ((exp(2.0 * corr_ci_lower) - 1.0) ./ (exp(2 * corr_ci_lower) + 1.0));
corr_ci_upper = ((exp(2.0 * corr_ci_upper) - 1.0) ./ (exp(2 * corr_ci_upper) + 1.0));

rmse_lower = rmses * sqrt(num_clips-4) / sqrt( chi2inv(0.025, num_clips-4));
rmse_upper = rmses * sqrt(num_clips-4) / sqrt( chi2inv(0.975, num_clips-4));

or_lower = outratios + 1.96 * sqrt( outratios .* (1.0 - outratios) ./ num_clips );
or_upper = outratios - 1.96 * sqrt( outratios .* (1.0 - outratios) ./ num_clips );
```

B.9 How to Compute Significant Differences Using RMSE

```
% Given the following variables:
% ‘rmse_a’ holds one model’s RMSE between VM and super-set DMOS.
% ‘rmse_b’ holds another model’s RMSE between VM and super-set DMOS.
%  AND where rmse_a > rmse_b (i.e., model A has a worse RMSE than model B)
% ‘num_clips_a’ holds the number of clips used to compute rmse_a
% ‘num_clips_b’ holds the number of clips used to compute rmse_b
% ‘df’ is the degrees of freedom, computed as when calculating RMSE in section B.7
% ‘ci’ is the confidence level: 0.95 for 95% confidence, 0.75 for 75% confidence, etc.
% The variable ‘is_same’ becomes 1 if the two models are statistically equivalent,
% and 0 if model A has significantly worse performance than model B.
%
% The function finv computes the inverse of the F cumulative distribution function and is found
% in the MATLAB Statistical Toolbox.

if (rmse_a.^2) / (rmse_b.^2) < finv(ci, num_clips_a-df, num_clips_b-df),
        is_same = 1;
else
        is_same = 0;
end
```

FORM **NTIA-29** U.S. DEPARTMENT OF COMMERCE
(4-80) NAT'L. TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION NO. TR-09-457	2. Government Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Techniques for Evaluating Objective Video Quality Models Using Overlapping Subjective Data Sets		5. Publication Date Nov. 2008 6. Performing Organization NTIA/ITS.T
7. AUTHOR(S) Margaret H. Pinson and Stephen Wolf		9. Project/Task/Work Unit No. 3141000-300
8. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Telecommunication Sciences National Telecommunications & Information Administration U.S. Department of Commerce 325 Broadway Boulder, CO 80305		10. Contract/Grant No.
11. Sponsoring Organization Name and Address National Telecommunications & Information Administration Herbert C. Hoover Building 14th & Constitution Ave., NW Washington, DC 20230		12. Type of Report and Period Covered

14. SUPPLEMENTARY NOTES

15. ABSTRACT

This report presents techniques for evaluating objective video quality models using overlapping subjective data sets. The techniques are demonstrated using data from the Video Quality Experts Group (VQEG) Multi-Media (MM) Phase I experiments. These results also provide a supplemental analysis of the performance achieved by the objective models that were submitted to the MM Phase I experiments. The analysis presented herein uses the subjective scores from the common set of video clips to map all the subjective scores from the 13 or 14 experiments (at a given image resolution) onto a single subjective scale. This mapping allows for more powerful analysis techniques to be performed. Resolving power values are presented for each model and resolution. On a per-clip level, models' responses to stimuli are analyzed with respect to all stimuli, to each coding algorithm, to coding-only impairments, and to transmission error impairments. The models' responses to stimuli are also analyzed on a per-system and per-scene level, which indicates the amount of improvement possible when averaging over multiple scenes or systems.

16. Key Words

Combining; correlation; mapping; multi-media; objective; performance; quality; subjective; video, VQEG

17. AVAILABILITY STATEMENT □ UNLIMITED.	18. Security Class. (This report) Unclassified	20. Number of pages
	19. Security Class. (This page) Unclassified	21. Price:

NTIA FORMAL PUBLICATION SERIES

NTIA MONOGRAPH (MG)

A scholarly, professionally oriented publication dealing with state-of-the-art research or an authoritative treatment of a broad area. Expected to have long-lasting value.

NTIA SPECIAL PUBLICATION (SP)

Conference proceedings, bibliographies, selected speeches, course and instructional materials, directories, and major studies mandated by Congress.

NTIA REPORT (TR)

Important contributions to existing knowledge of less breadth than a monograph, such as results of completed projects and major activities. Subsets of this series include:

NTIA RESTRICTED REPORT (RR)

Contributions that are limited in distribution because of national security classification or Departmental constraints.

NTIA CONTRACTOR REPORT (CR)

Information generated under an NTIA contract or grant, written by the contractor, and considered an important contribution to existing knowledge.

JOINT NTIA/OTHER-AGENCY REPORT (JR)

This report receives both local NTIA and other agency review. Both agencies' logos and report series numbering appear on the cover.

NTIA SOFTWARE & DATA PRODUCTS (SD)

Software such as programs, test data, and sound/video files. This series can be used to transfer technology to U.S. industry.

NTIA HANDBOOK (HB)

Information pertaining to technical procedures, reference and data guides, and formal user's manuals that are expected to be pertinent for a long time.

NTIA TECHNICAL MEMORANDUM (TM)

Technical information typically of less breadth than an NTIA Report. The series includes data, preliminary project results, and information for a specific, limited audience.

For information about NTIA publications, contact the NTIA/ITS Technical Publications Office at 325 Broadway, Boulder, CO, 80305 Tel. (303) 497-3572 or e-mail info@its.bldrdoc.gov.

This report is for sale by the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161,Tel. (800) 553-6847.